



# Combinatoire des fonctions de parking : espèces, énumération d'automates et algèbres de Hopf

Jean-Baptiste Priez

## ► To cite this version:

Jean-Baptiste Priez. Combinatoire des fonctions de parking : espèces, énumération d'automates et algèbres de Hopf. Combinatoire [math.CO]. Université Paris Saclay (COMUE), 2015. Français. NNT : 2015SACLS111 . tel-01274077

**HAL Id: tel-01274077**

**<https://theses.hal.science/tel-01274077>**

Submitted on 15 Feb 2016

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

NNT : 2015SACLS111

THÈSE DE DOCTORAT  
DE  
L'UNIVERSITÉ PARIS-SACLAY  
PRÉPARÉE À  
L'UNIVERSITÉ PARIS-SUD

ECOLE DOCTORALE N° 580  
Sciences et technologies de l'information et de la communication  
Spécialité Mathématiques et informatique

Par

**M. Jean-Baptiste Priez**

Combinatoire des fonctions de parking :  
Espèces, Énumération d'automates et Algèbres de Hopf

**Thèse présentée et soutenue à Orsay, le lundi 7 décembre 2015 :**

**Composition du Jury :**

M. Florent Hivert	Professeur de l'Université Paris-Sud	Directeur de thèse
M. Jean-Christophe Aval	Chargé de recherche à l'Université de Bordeaux	Rapporteur
M. Jean-Gabriel Luque	Professeur de l'Université de Rouen	Rapporteur
M. Sylvain Conchon	Professeur de l'Université Paris-Sud	Président
Mme Sylvie Corteel	Directrice de recherche CNRS à l'Université Paris-Diderot	Examinatrice
M. Loïc Foissy	Professeur de l'Université du Littoral Côte d'Opale	Examineur
M. Cyril Nicaud	Professeur de l'Université Paris-Est	Examineur

## Résumé

Cette thèse se situe dans les domaines de la combinatoire algébrique, bijective et énumérative. Elle s'intéresse à l'étude des fonctions de parking généralisées suivant ces trois axes.

Dans une première partie, on s'intéresse aux fonctions de parking généralisées en tant qu'espèce de structures combinatoires (théorie introduite par A. JOYAL et développée F. BERGERON, G. LABELLE et P. LEROUX). On définit cette espèce à partir d'une équation fonctionnelle faisant intervenir l'espèce des séquences d'ensembles. On obtient un relèvement non-commutatif de la série indicatrice de cycles dans les fonctions symétriques non-commutatives, exprimé dans différentes bases. Par spécialisation, on obtient de nouvelles formules d'énumérations des fonctions de parking généralisées et de leurs types d'isomorphismes.

En remplaçant l'espèce des ensembles par d'autres espèces dans l'équation fonctionnelle, on définit de nouvelles structures : les  $\chi$ -tables de parking. Dans les cas particuliers où  $\chi : m \mapsto a + b(m - 1)$ , on établit une bijection entre les  $\chi$ -tables de parking et de nouvelles structures arborescentes, généralisant la bijection de C. H. YAN entre les  $\chi$ -fonctions de parking et les séquences de  $a$  forêts de  $b$ -arbres.

Dans une seconde partie, on s'intéresse à l'énumération d'automates. On commence par construire une bijection simple entre les automates (non-initiaux) et les séquences d'ensembles. À partir de cette bijection, on extrait la sous-famille des automates quasi-distingués (c'est-à-dire les automates pour lesquels les couples status de terminaison et fonction de transition des états sont distincts). L'énumération de ces automates quasi-distingués fournit une meilleure borne supérieure pour le nombre d'automates minimaux que celle obtenue par M. DOMARATZKI & al.

Ensuite, on construit une nouvelle bijection entre les  $2m^k$ -fonctions de parking et les automates acycliques (non-initiaux) sur un alphabet à  $k$  symboles. De cette dernière, on extrait, directement sur les fonctions parking, de nombreuses informations de structure sur les automates, en particulier des informations liées à la minimalité. À partir de ces informations, on déduit une formule d'énumération des automates acycliques minimaux.

Dans une troisième partie, on formalise la technique commune de réalisation polynomiale des algèbres de Hopf : **FQSym**, **WQSym**, **PQSym**, etc.. Pour ceci, on définit la notion de type d'alphabet et d'application partitionnante. La notion d'application partitionnante formalise les bonnes propriétés de la standardisation, le tassement, la parkisation, etc. associées à ces précédentes algèbres de Hopf. On montre que certaines opérations, produit cartésien, coloration, union ou encore intersection, stabilisent ces notions. À partir de celles-ci, on définit deux constructions d'algèbres de Hopf combinatoire en dualité ; et l'on montre qu'elles sont automatiquement munies de structures d'algèbres dendriformes et du produit  $\#$ .

En guise d'applications, on définit, pour toute famille de  $\chi$ -fonctions de parking, une application généralisant la parkisation. On montre que cette dernière est une application partitionnante si et seulement si  $\chi : n \mapsto 1 + m(n - 1)$ . Ceci permet de retrouver les algèbres de Hopf sur les  $m$ -fonctions de parking généralisées de J.-C. NOVELLI et J.-Y. THIBON.

**Mots-clés :** combinatoire algébrique, bijective, énumérative ; fonction de parking généralisé ; automate acyclique ; algèbre de Hopf ; fonction symétrique non-commutative ; réalisation polynomiale

## Abstract

This thesis comes within the scope of algebraic, bijective and enumerative combinatorics. It deals with the study of generalized parking functions following those axes.

In the first part, we are interested in generalized parking as a species of combinatorial structures (theory introduced by A. JOYAL and developed by F. BERGERON, G. LABELLE et P. LEROUX). We define this species from a functional equation involving the species of set sequences. We lift the cycle index serie to the non-commutative symmetric functions, expressed in several bases. By specialization, we obtain new enumeration formula for generalized parking and its isomorphism types.

By replacing the species of sets by others species in the functional equation, one defines new structures : the  $\chi$ -parking tables. In particular cases with  $\chi : m \mapsto a + b(m - 1)$ , we define a bijection between the  $\chi$ -parking tables and new tree structures. This gives a generalization of the C. H. YAN bijection between  $\chi$ -parking functions and the sequences of  $a$ -forest of  $b$ -trees.

In the second part, we are interested in the enumeration of automata. Firstly, we construct a simple bijection between (non-initial) automata and sequences of sets. From this bijection we extract a subfamily of quasi-distinguished automata (this means the automata which all states have distinct accepting status and transition function). We obtain a better upper bound of the number of minimal automata than the one of M. DOMARATZKI & *al*.

Then we construct a new bijection between  $2m^k$ -parking functions and (non-initial) acyclic automata over an alphabet of  $k$  symbols. From this bijection we extract, from parking function, informations about minimality of automata. We deduce an enumeration formula of the minimal acyclic automata.

In a third part, we formalize the common technique of polynomial realization of Hopf algebras : **FQSym**, **WQSym**, **PQSym**, *etc.*. We define a notion of type of alphabet and partitioning map. We highlight some operation which stabilizes these notions. Based on this, we define two constructions of dual combinatorial Hopf algebra ; and we show that they are automatically endowed of dendriform coalgebra, and  $\#$ -product.

As an application, we define, for every family of  $\chi$ -parking functions, a generalization of the parkization. We show that this is a partitionning map if and only if  $\chi : n \mapsto 1 + m(n - 1)$ . From this we obtain the  $m$ -**PQSym** Hopf algebras of J.-C. NOVELLI et J.-Y. THIBON.

**Keywords :** algebraic, bijective, enumerative combinatorics ; generalized parking function ; acyclic automata ; Hopf algebra ; non-commutative symmetric function ; polynomial realization

# Remerciements

Allez savoir pourquoi, les britanniques ne m'ont jamais autant fait marrer que ces trois dernières années...

I feel a very unusual sensation – if it is not indigestion, I think it must be gratitude.  
(Benjamin DISRAELI)

Je tiens à commencer par remercier celui grâce à qui cette aventure a commencé, Florent HIVERT, mon directeur de thèse. Après m'avoir enseigné le **Pascal** en première année de licence à l'Université de Rouen ; lui, Éric LAUGEROTTE et Jean-Gabriel LUQUE m'ont fait découvrir ce domaine passionnant de la combinatoire. Cette passion m'a poussé à suivre Florent au L.R.I. pour mon stage de master et la thèse ensuite. Je le remercie chaleureusement de sa patience, sa générosité et sa bienveillance. Merci de m'avoir fait confiance et de m'avoir guidé avec attention jusqu'à ce moment !

Je remercie mes rapporteurs, Jean-Christophe AVAL et Jean-Gabriel LUQUE, pour tout le temps qu'ils m'ont consacré en lisant et commentant ce manuscrit. Je remercie aussi les examinateurs, Sylvain CONCHON, Sylvie CORTEEL, Loïc FOISSY et Cyril NICAUD, d'avoir accepté d'être mon jury. En particulier, merci à Sylvie et Jean-Christophe pour leurs invitations à partager mon travail à leurs séminaires respectifs et leurs gentilleses pendant ces échanges ; merci à Sylvain de m'avoir aiguillé et soutenu pendant mes premières années de monitorat ; merci à Loïc pour ses précieuses explications et conseils prodigués au S.L.C. ; merci à Cyril pour ses recommandations ainsi que pour les nombreux échanges par mails ou à Marnes que nous avons eus.

Je remercie aussi chaleureusement Nicolas M. THIÉRY. Je ne compte plus le nombre d'heures qu'il m'a consacré à m'expliquer des maths ou du SAGE. Il m'a fait confiance pour différents enseignements ; sa grande pédagogie ainsi que son management positif m'inspirent un profond respect et la volonté de suivre ses pas.

Pendant ces trois années de thèse, j'ai eu le plaisir de participer à de nombreuses fois au séminaire du vendredi à Marne-la-Vallée. Je remercie Jean-Christophe NOVELLI et Jean-Yves THIBON pour leur accueil et leurs précieux conseils et recommandations. Un grand merci à Jean-Christophe qui m'a orienté vers l'exploration des fonctions de parking ; je le remercie aussi pour les nombreux moments qu'il m'a accordés dans son bureau. Je souhaite remercier aussi Viviane PONS (ex-Marneuse), Nicolas BORIE, Samuele GIRAUDO et Matthieu JOSUAT-VERGÈS pour leur gentillesse et leur bienveillance à mon égard. Nos discussions – scientifique ou non – m'ont beaucoup apporté et ils sont assurés de mon amitié sincère ! Il en va de même pour les jeunes du bureau des thésards : Arthur, Grégory, Olivier, Vincent et Zakaria ; merci pour tous les bons moments à Marnes, Ellwangen, Bertinoro ou même Daejeon. Merci aussi à toute l'équipe du « troisième » pour tous les bons petits plats et les cours de cuisine.

Grâce au financement du L.I.R.C.O., j'ai eu l'opportunité de partir à Montréal rencontrer François BERGERON. Son enseignement de la théorie des espèces, pendant la petite école d'été et dans son bureau entre quatre yeux, m'a apporté plein de nouvelles idées. Je le remercie pour ce séjour et tous ces précieux conseils ! Ce séjour m'a permis de ren-

contrer des personnes formidables : Amy, Johanne, Alejandro, Franco, Luis, Mathieu, Nathan, Vivien et Yannic. Je vous remercie tous pour votre accueil chaleureux.

Je remercie également Mike ZABROCKI pour son super accueil à Toronto (et ses délicieux Gin Tonic); Marcelo AGUIAR et Aaron LAUVE pour m'avoir invité et si bien reçu à Cornell et Loyola.

Je suis aussi reconnaissant à toute l'équipe de l'Université de Rouen pour m'avoir encadré et soutenu de la L1 jusqu'au bout du master I.T.A.!

Ces trois années m'ont permis de faire de fantastiques rencontres : Ștefania, Aladin et Nathann. Je suis heureux d'avoir traversé cette aventure avec eux et j'espère avoir l'occasion d'en vivre de nouvelles en leurs compagnies. Vous êtes des amis précieux!

J'ai aussi une pensée pour toutes les autres belles rencontres : Bérénice, Élie, Joël, Louis, Thibault, Vincent, Yannis, etc..

Merci les amis : Claudia, Ophélie, Antoine, Danil, Étienne, Édouard, Florian, François, Frédéric, Guillaume, Julien, Sébastien et Steeve!!! Merci d'être là!! Merci de ne pas m'avoir laissé devenir autiste! Je vous aime.

Pour finir, je remercie ma Clémence! Elle est là pour moi et me comble au quotidien! Je remercie ma famille et celle de Clémence! Je n'échangerai tout ça pour rien au monde!

# Table des matières

<b>Introduction</b>	<b>10</b>
<b>1. Rappels et notations</b>	<b>17</b>
1.1. Quelques classes combinatoires . . . . .	17
1.1.1. Partitions d'entiers . . . . .	18
1.1.2. Partitions d'ensembles . . . . .	19
1.1.3. Compositions d'entiers . . . . .	20
1.1.4. Ordres linéaires . . . . .	22
1.1.5. Permutations et groupe symétrique . . . . .	22
1.1.6. Mots et monoïdes libres . . . . .	23
1.1.7. Relations d'équivalences . . . . .	25
1.2. Fonctions symétriques . . . . .	26
1.2.1. Fonctions monomiales . . . . .	26
1.2.2. Sommes de puissances . . . . .	26
1.2.3. Fonction élémentaires . . . . .	27
1.2.4. Fonctions homogènes . . . . .	27
1.2.5. Quelques égalités . . . . .	27
1.2.6. $Sym_n$ et spécialisations . . . . .	28
1.3. Fonctions symétriques non-commutatives . . . . .	29
1.3.1. Bases . . . . .	29
1.3.2. Produit interne . . . . .	31
1.3.3. Opérateurs d'Adams . . . . .	32
1.3.4. Base monomiale des fonctions quasi-symétriques . . . . .	32
1.3.5. Transformation d'alphabet . . . . .	33
1.4. Théorème de dénombrement de REDFIELD-PÓLYA . . . . .	34
1.4.1. Notions de théorie des groupes . . . . .	34
1.4.2. Lemme de Burnside . . . . .	35
1.4.3. Théorème de REDFIELD-PÓLYA . . . . .	36
1.5. Théorie des espèces . . . . .	37
1.5.1. Définition . . . . .	37
1.5.2. Séries associées . . . . .	39
1.5.3. Espèces caractéristiques . . . . .	42
1.5.4. Somme, Produit, Composition et Équation fonctionnelle . . . . .	43

<b>1. Autour des fonctions de parking</b>	<b>47</b>
<b>2. Espèce des séquences finies d'ensembles</b>	<b>50</b>
2.1. Séquences et chemins décorés . . . . .	50
2.2. Types d'isomorphismes . . . . .	51
2.3. Séries génératrices . . . . .	52
2.4. Relèvement non-commutatif . . . . .	53
2.5. Interprétation de la base des fonctions complètes . . . . .	53
2.6. Interprétation de la base des rubans . . . . .	54
2.7. Interprétation de la base des fonctions élémentaires . . . . .	55
2.8. Formule de $q$ -énumération . . . . .	55
<b>3. Fonctions de parking</b>	<b>57</b>
3.1. Espèce des fonctions de parking $\mathcal{P}$ . . . . .	57
3.2. Séquences d'ensembles et fonctions de parking . . . . .	57
3.3. $\mathcal{P}$ -structures . . . . .	58
3.3.1. Fonctions entières . . . . .	59
3.3.2. Séquences d'ensembles . . . . .	59
3.3.3. Mots . . . . .	59
3.3.4. Chemins de Dyck décorés . . . . .	60
3.4. Série génératrice . . . . .	62
3.5. Série indicatrice de cycles . . . . .	63
3.5.1. $q$ -analogues . . . . .	63
<b>4. Fonctions de parking généralisées</b>	<b>65</b>
4.1. Définition récursive . . . . .	66
4.1.1. Intuition sur les chemins décorés . . . . .	66
4.1.2. Grammaire récursive . . . . .	69
4.2. Caractère de Frobenius non-commutatif . . . . .	71
4.2.1. Fonctions complètes . . . . .	71
4.2.2. Développement des opérateurs d'Adams . . . . .	74
4.3. Caractères dans d'autres bases . . . . .	77
4.3.1. Fonctions rubans . . . . .	77
4.3.2. Fonctions élémentaires . . . . .	78
4.4. Formules d'énumération . . . . .	79
4.4.1. Énumération de $\chi$ -fonctions de parking . . . . .	80
4.4.2. Énumération de $\chi$ -chemins . . . . .	81
<b>5. Tables de parking et arborescences</b>	<b>83</b>
5.1. Tables de parking (généralisées) . . . . .	83
5.1.1. Définition . . . . .	84
5.1.2. Série indicatrice de cycles . . . . .	84
5.1.3. Formules d'énumération . . . . .	87
5.1.4. Cas particulier : identité . . . . .	87



5.2.	Forêts d'arbres enracinés . . . . .	87
5.2.1.	Énumération des forêts d'arbres enracinés . . . . .	89
5.2.2.	Codage de Prüfer . . . . .	91
5.3.	Bijection FOATA-RIORDAN-YAN . . . . .	95
5.3.1.	Structures arborescentes . . . . .	95
5.3.2.	Arborescences vers tables de parking . . . . .	98
5.3.3.	Tables de parking vers arborescences . . . . .	102
5.3.4.	Énumération . . . . .	105

## **II. Énumération d'automates 108**

### **6. Théorie des langages et automates 110**

6.1.	Langages rationnels . . . . .	110
6.2.	Automates . . . . .	111
6.3.	Automates minimaux . . . . .	113
6.4.	Automates acycliques et langages finis . . . . .	115
6.5.	Automates acycliques minimaux . . . . .	116

### **7. Énumération d'automates 119**

7.1.	Énumération d'automates accessibles . . . . .	119
7.2.	Énumération d'automates quasi-distingués . . . . .	122
7.2.1.	Expérimentation : série indicatrice de cycles des automates (non-initiaux) distingués . . . . .	123
7.3.	Énumération d'automates quasi-distingués accessibles . . . . .	124

### **8. Énumération d'automates acycliques minimaux 126**

8.1.	Bijection entre les automates acycliques (non-initiaux) et les fonctions de parking généralisées . . . . .	127
8.1.1.	Ordre de parking et ordre alphabétique . . . . .	128
8.1.2.	Factorisation de parking et fonctions de transition . . . . .	129
8.1.3.	Fonction de parking vers automate acyclique . . . . .	132
8.2.	Co-accessibilité . . . . .	136
8.3.	Quasi-distinguabilité et distinguabilité . . . . .	139
8.3.1.	Quasi-distingués . . . . .	140
8.3.2.	Distingués : co-accessibles et quasi-distingués . . . . .	141
8.4.	Automates acycliques étendus . . . . .	142
8.4.1.	Automates étendus co-accessible . . . . .	145
8.4.2.	Automates étendus quasi-distingués . . . . .	145
8.4.3.	Automates étendus avec contrainte . . . . .	146
8.5.	Automates acycliques minimaux . . . . .	147

### III. Constructions d'algèbres de Hopf combinatoires 149

#### 9. Quelques prérequis 153

9.1. Algèbre libre . . . . .	153
9.1.1. Algèbres libres . . . . .	153
9.1.2. Quotient commutatif d'algèbres libres . . . . .	154
9.1.3. Algèbres libres et produit tensoriels complétés . . . . .	154
9.2. Algèbre de Hopf . . . . .	155
9.2.1. Définition . . . . .	155
9.2.2. Axiomes et diagrammes . . . . .	155
9.2.3. Algèbre de Hopf combinatoire . . . . .	158
9.3. Catégorie . . . . .	159
9.3.1. Catégories . . . . .	159
9.3.2. Foncteurs . . . . .	160
9.3.3. Transformations naturelles . . . . .	160
9.3.4. Catégorie magmatique . . . . .	161

#### 10. Réalisation polynomiale 163

10.1. Type d'alphabet . . . . .	163
10.1.1. Exemples . . . . .	164
10.1.2. Opérations . . . . .	165
10.2. Application partitionnante . . . . .	166
10.2.1. Opérations . . . . .	169
10.3. Algèbre de Hopf $\mathcal{H}_\phi$ . . . . .	172
10.3.1. Espace vectoriel . . . . .	173
10.3.2. Algèbre . . . . .	173
10.3.3. Coalgèbre . . . . .	177
10.3.4. Bialgèbre et algèbre de Hopf . . . . .	182
10.4. Algèbre de Hopf $\mathcal{H}_\phi^\vee$ . . . . .	183
10.4.1. Espace vectoriel . . . . .	183
10.4.2. Algèbre quotient . . . . .	184
10.4.3. Coalgèbre . . . . .	186
10.4.4. Algèbre de Hopf . . . . .	187
10.4.5. Dualité entre $\mathcal{H}_\phi$ et $\mathcal{H}_\phi^\vee$ . . . . .	189

#### 11. Extensions 191

11.1. Quotient commutatif . . . . .	191
11.2. Réalisation de dièse-algèbres . . . . .	192
11.3. Réalisation dendriforme . . . . .	194
11.3.1. Définitions . . . . .	194
11.3.2. Coalgèbre dendriforme . . . . .	196
11.3.3. Bialgèbre codendriforme . . . . .	198
11.3.4. Bialgèbre bidendriforme . . . . .	199

<b>12. Exemples de réalisations</b>	<b>203</b>
12.1. <b>FQSym</b> . . . . .	203
12.1.1. Alphabets totalement ordonnés et standardisation . . . . .	203
12.1.2. <b>FQSym</b> duale, alias $\mathcal{H}_{\text{std}}$ . . . . .	205
12.1.3. <b>FQSym</b> , alias $\mathcal{H}_{\text{std}}^{\vee}$ . . . . .	206
12.1.4. Propriété dendriforme . . . . .	207
12.2. <b><math>m</math>-PQSym</b> . . . . .	207
12.2.1. Parkisation . . . . .	208
12.2.2. Application partitionnante . . . . .	210
<b>Perspectives</b>	<b>220</b>
<b>A. Développement Sage</b>	<b>223</b>
<b>Bibliographie</b>	<b>225</b>
<b>Nomenclature</b>	<b>233</b>
<b>Index</b>	<b>235</b>

# Introduction

There is a theory which states that if ever anyone discovers exactly what the Universe is for and why it is here, it will instantly disappear and be replaced by something even more bizarre and inexplicable.

There is another theory which states that this has already happened.

(Douglas ADAMS, *The Restaurant at the End of the Universe*)

Loin de l'idée de comprendre l'Univers et le pourquoi de tout cela, ce manuscrit est un guide pour tout explorateur qui souhaite se familiariser avec différents aspects de la combinatoire des *fonctions de parking*.

D'un point de vue général, la *Combinatoire* est l'étude des collections finies d'objets. Elle se divise en plusieurs tendances ; parmi les principales on trouve la *combinatoire énumérative* qui s'intéresse au dénombrement de ces collections ; la *combinatoire algébrique* qui enrichit ces collections de structures algébriques pour en étudier la combinatoire ; la *combinatoire bijective* qui met en relation différentes collections ; et, la *combinatoire analytique* qui étudie les *séries génératrices* en utilisant l'analyse complexe.

De mon point de vue, la *Combinatoire* est un art, une passerelle vers l'ensemble des domaines des sciences : *mathématiques, informatique, physique, biologie, économie, etc..* J'aime à croire que cet art traduit des problèmes importants et souvent complexes de manière ludique et récréative :

Combien de couples (de lapins) obtient-on en un an si chaque couple engendre tous les mois un nouveau couple à compter du troisième mois de son existence ?  
(Leonardo FIBONACCI, *Liber abaci*)

À partir de ce *problème combinatoire* élémentaire, faisant naître la suite de Fibonacci, on introduit le nombre d'or et ses multiples applications : *architecture, biologie* ou encore *minéralogie*. Cet aspect omniprésent de vulgarisation est, pour moi, le cœur de ce domaine. Il permet d'aborder les problèmes sous différents angles, d'améliorer leur compréhension, et de les transférer petit à petit, du monde de la Recherche, au monde de l'Enseignement, et enfin, de les traduire en des applications au monde de l'Entreprise.

## Fonctions de parking

Les *fonctions de parking*, introduites par Alan G. KONHEIM et Benjamin WEISS [KW66], sont issues d'un *problème informatique* lié aux fonctions de hachage, et apparaissent maintenant dans de nombreux autres domaines combinatoires. Soit  $U$  un ensemble fini à  $n$  éléments. Une fonction de parking est une fonction  $f : U \rightarrow \mathbb{N}_+$  telle que

$$\#\{f(u) \leq k \mid u \in U\} \geq k, \quad \text{pour tout } i \leq k \leq n. \quad (1.23)$$

Par exemple voici les premières fonctions de parking pour  $U = \emptyset, \{1\}, \{1, 2\}$  et  $\{1, 2, 3\}$  :

$\epsilon, 1, 11, 12, 21, 111, 112, 121, 211, 113, 131, 311, 122, 212, 221, 123, 132, 213, 231, 312, 321,$

où l'on note  $f$  par le mot  $f(1)f(2)f(3)$ . Ces fonctions sont en bijection avec les forêts d'arbres enracinés ou encore les séquences de Prüfer. Elles apparaissent en *géométrie* : on les trouve dans l'étude de certains polytopes [PS02] et elles indexent les régions de l'arrangement d'hyperplans de Jian Yi SHI [Shi86, Chapitre 7]. On les voit apparaître dans l'étude des polynômes de Gončarov [KY03 ; KSY06], utile en interpolation et en *analyse numérique*. En particulier, Joseph P. S. KUNG et Catherine H. YAN [KY03] déduisent de ces polynômes l'élégante formule d'énumération :

$$p(\chi; n) = \sum_{s=1}^n (-1)^{s-1} \binom{n}{s} \chi(n-s-1)^s p(\chi; n-s). \quad [\text{KY03, Théorème 4.2}]$$

En combinatoire algébrique, plusieurs structures d'algèbres de Hopf sont définies sur les fonctions de parking [NT04a ; NT14]. Ces fonctions jouent aussi un rôle important en *théorie des représentations* ; en particulier dans la fameuse conjecture/théorème  $n!$ , à propos de la dimension du module des polynômes harmoniques diagonaux [GH96 ; Hai94]. Tout ceci en fait un objet majeur de la combinatoire. En outre, dans ce manuscrit, nous allons relier ces structures à un nouvel autre domaine : la *théorie des automates* [Pri15b]. En effet, nous allons définir une bijection entre les  $2m^k$ -fonctions de parking et les *automates acycliques (non-initiaux) finis déterministes* (sur un alphabet de  $k$  symboles).

## Automates

Les automates sont des outils théoriques d'une importance capitale en informatique. Ils apparaissent dans différents composants des ordinateurs comme les unités logiques et arithmétiques des microprocesseurs. Pouvoir tirer au hasard ces structures permet de tester des conjectures et calculer des complexités de programmes réels. L'obtention d'une formule d'énumération exacte permet très souvent d'obtenir un algorithme de tirage aléatoire uniforme. Ces diverses applications rendent donc le problème de l'énumération des automates, en particulier des automates minimaux, très intéressant. Plusieurs équipes ont ainsi travaillé dessus ces quinze dernières années [CH04 ; Cal08 ; DKS02 ; Dom03 ; Dom04 ; AMR07 ; AMR08]. En 2006, Valery A. LISKOVETS résout le problème de l'énumération des automates acycliques dans [Lis06]. Il obtient la formule suivante :

$$a(k; n) = \sum_{s=1}^n (-1)^{s-1} \binom{n}{s} 2^s (n-s+1)^{ks} a(k; n-s), \quad [\text{Lis06, Théorème 3.1}]$$

pour l'énumération des automates acycliques (non-initiaux). Marco ALMEIDA, Nelma MOREIRA et Rogério REIS au Portugal, ou encore Cezar CÂMPEANU, Wing H. Ho, Michael DOMARATZKI, Derek KISMAN, Jeffrey O. SHALLIT ou David CALLAN, en Amérique du Nord se

sont intéressés au problème de l'énumération des *automates minimaux acycliques*. C'est-à-dire à l'énumération des automates minimaux reconnaissant des langages finis. Ce problème de l'énumération des automates acycliques minimaux est resté ouvert jusqu'à mon article [Pri15b].

## Théorie des espèces

Pendant longtemps, la combinatoire a manqué d'outils génériques, résolvant chaque problème au cas par cas. La combinatoire symbolique est un outil très puissant dont l'idée centrale est de décrire les ensembles d'objets combinatoires à l'aide de grammaire récursive. On peut ensuite automatiquement obtenir des formules d'énumérations [FS09]. Lorsque les objets possèdent des symétries, le bon cadre combinatoire est la *théorie des espèces* [BLL98].

Cette théorie

permet d'expliquer, de façon élégante, l'utilisation de calcul de séries génératrices pour l'énumération de structures. [...]

D'une manière unifiée, elle permet autant de comprendre l'utilisation de séries génératrices pour l'énumération de structures étiquetées ou non, que l'élaboration d'outils d'analyse et de spécification de ces structures. Ainsi, elle transforme en équations fonctionnelles (ou différentielles) la spécification récursive de structures arborescentes. Englobant la description de structures avec la description de l'action de groupes de permutations sur ces structures, la théorie des espèces unifie le calcul de fonctions génératrices avec la théorie de Pólya.

(François BERGERON, [page web](#))

La *théorie de Pólya* [Red27 ; Pól37] utilise l'algèbre de Hopf des fonctions symétriques pour encoder ces séries indicatrices de cycles qui unifient le « calcul de fonctions génératrices ». De manière générale, les algèbres de Hopf « combinatoires » fournissent des outils puissants pour résoudre les problèmes d'énumération. Elles améliorent l'expressivité des séries génératrices en relevant des informations combinatoires au niveau algébrique. En particulier, les notions de produit et de coproduit s'expriment, en termes combinatoires, comme un *assemblage* et un *désassemblage* de structures.

## Algèbres de Hopf

L'algèbre de Hopf des fonctions symétriques, indexée par les partitions d'entiers, peut se définir à partir de réalisations polynomiales dans l'algèbre des séries (graduées) en une infinité de variables commutatives  $\mathbb{X} = x_1 + x_2 + x_3 + \dots$ . En utilisant ces réalisations, l'assemblage des partitions s'exprime alors comme le produit de concaténation des « mots » des séries, et le désassemblage correspond à une extraction d'un ensemble de variables. Le point intéressant de ces réalisations est que la justification de la structure d'algèbre de Hopf (produit associatif, coproduit coassociatif, etc.) est presque gratuite. Cet aspect algébrique est traité au niveau des séries de mots et la compréhension des opérations d'assemblage et de désassemblage est purifiée de toute

contrainte algébrique. Ces opérations peuvent ainsi être étudiées d'un pur point de vue combinatoire et algorithmique.

Ce concept de réalisation polynomiale d'algèbres de Hopf a été largement développé, et intégré dans le folklore de la combinatoire algébrique, par le *phalanstère de combinatoire de Marne-la-Vallée* auprès de Jean-Yves THIBON [KLT97 ; NT06, etc. ]. Dans le manuscrit, on donne un formalisme détaillé de cette théorie.

## Problématique

Il est frappant de remarquer que la formule de Valery A. LISKOVETS [Lis06, Théorème 3.1] est un cas particulier de celle de Joseph P. S. KUNG et Catherine H. YAN [KY03, Théorème 4.2] pour  $\chi : m \mapsto 2m^k$ . Partant de ce constat, la motivation principale de ce travail est de fournir une explication à cette coïncidence.

Le point de départ est le suivant : les fonctions de parking généralisées se définissent, en termes d'espèces de structures [BLL98], récursivement par l'équation fonctionnelle :

$$\mathcal{P}(\chi) = (\mathbf{E}^{\chi(1)})_0 + \sum_{n \geq 1} (\mathbf{E}^{\chi(1)})_n \cdot \mathcal{P}(\psi_n) \quad (4.3 + \text{Lemme 4.1.6})$$

avec  $\psi_n : m \mapsto \chi(n + m) - \chi(1)$ . Cette équation traduit les  $\chi$ -fonctions de parking en termes de séquences d'ensembles. La série indicatrice de cycles obtenue automatiquement, à partir de la théorie des espèces, se relève naturellement dans les fonctions symétriques non-commutatives comme le caractère de Frobenius non-commutatif pour l'action de l'algèbre de Hecke dégénérée à  $q = 0$ .

L'expression récursive des  $2m^k$ -fonctions de parking, en terme de séquences d'ensembles, permet de définir et étudier une bijection avec les automates acycliques (non-initiaux sur un alphabet à  $k$  symboles). Au travers de cette bijection, de nombreuses informations sur la structure d'automate sont immédiatement lisibles sur les fonctions de parking. Grossièrement, le découpage récursif d'une  $\chi$ -fonction de parking traduit une notion de hauteur dans l'automate, et les étiquettes d'un même ensemble de la séquence encodent l'ensemble des états ayant le même et unique langage droit. Ces interprétations combinatoires, couplées à l'expressivité des formules d'énumérations des fonctions de parking, permettent de déduire une formule d'énumération exacte des automates acycliques minimaux.

Les différents caractères obtenus s'expriment dans des algèbres de Hopf, souvent dites combinatoires. Ces algèbres combinatoires peuvent généralement se réaliser polynomialement dans une algèbre libre à partir des techniques développées par le *phalanstère de combinatoire de Marne-la-Vallée*. Ces techniques font partie du folklore de la combinatoire algébrique depuis une vingtaine d'années. Une seconde motivation de ce travail est d'exprimer ces techniques dans une théorie unifiée.

À partir de la théorie des réalisations polynomiales d'algèbres de Hopf, développée ici, nous investiguerons les algèbres de Hopf  $m$ -PQSym de Jean-Christophe NOVELLI et

Jean-Yves THIBON [NT14]. Nous tenterons de généraliser l'application partitionnante de parkisation.

Ce manuscrit se découpe en trois parties où sont développées chacune des idées énoncées dans cette introduction. Voici un résumé des trois grandes parties abordées :

## Fonctions de parking généralisées

Après avoir introduit quelques définitions et notations de structures combinatoires (partitions, compositions, permutations, etc.) ainsi que quelques outils (Sym, NCSF, la théorie des espèces), dans la première partie, on s'intéresse aux *fonctions de parking généralisées* en tant qu'*espèce* de structures combinatoires (Partie I). Cette partie étend l'article [PV15] conjoint avec Aladin VIRMAUX et le travail non publié [Pri15a].

On définit cette espèce à partir d'une *équation fonctionnelle* engendrée par l'espèce des séquences d'ensembles (4.3). On déduit automatiquement de cette équation fonctionnelle une série indicatrice de cycles. Cette série se relève naturellement dans les *fonctions symétriques non-commutatives* comme le caractère de Frobenius de l'action de l'algèbre Hecke dégénérée en  $q = 0$  (4.4). Ce relèvement prolonge le travail de Jean-Christophe NOVELLI et Jean-Yves THIBON sur le caractère de Frobenius non-commutatif des fonctions de parking usuelles [NT08]. On exprime ce caractère dans les différentes bases des fonctions symétriques non-commutatives : homogènes (Théorème 4.2.7), rubans (Corollaire 4.3.5) et élémentaires (Théorème 4.3.7). La spécialisation de ces formules permet d'obtenir de nouvelles formules d'énumérations des fonctions de parking généralisées (§ 4.4.1) et de leurs types d'isomorphismes (§ 4.4.2).

En remplaçant, dans l'équation fonctionnelle, l'espèce des ensembles par d'autres espèces  $\mathbf{F}$ , on obtient de nouvelles structures de types parking : les  $\chi$ -tables de parking en  $\mathbf{F}$  (5.1). Par spécialisation de l'équation fonctionnelle, on déduit encore une fois des formules d'énumérations (Proposition 5.1.7 et § 5.1.3). Il est bien connu que les fonctions de parking (usuelles) sont en bijection avec les arbres enracinés, parmi les nombreuses bijections existantes, on peut citer les travaux de Dominique FOATA et John RIORDAN [FR74]. En se basant sur leur bijection, Catherine H. YAN étend la bijection aux  $a + b(m - 1)$ -fonctions de parking en bijection avec une famille de structures arborescentes [Yan01]. On étend la bijection de FOATA-RIORDAN-YAN aux  $a + b(m - 1)$ -tables de parking en  $\mathbf{F}$  en bijection avec de nouvelles structures arborescentes paramétrées par  $\mathbf{F}$  (§ 5.3). En guise d'application, on déduit que l'espèce des  $m$ -tables de parking en l'espèce des listes et en bijection avec l'espèce des arbres binaires (Exemple 5.1.5).

## Énumération d'automates

Dans une seconde partie (Partie II), on s'intéresse à l'énumération d'automates. On s'intéresse particulièrement au problème ouvert de l'énumération des *automates minimaux*. Dans une première sous-partie, on retrouve des résultats classiques d'énumération d'automates à partir d'une bijection simple avec des séquences d'ensembles (Chapitre 7). À partir de cette bijection, on extrait une sous-famille d'automates, dits *quasi-distingués* (§ 7.2). L'énumération de cette sous-famille permet d'obtenir une meilleure



borne supérieure pour l'énumération des automates minimaux que celle fournie par Mike DOMARATZKI, Derek KISMAN Jeffrey O. SHALLIT [DKS02].

Dans une seconde sous-partie (Chapitre 8), on fait un lien entre les formules d'énumération de Joseph P. S. KUNG et Catherine H. YAN [KY03] et celles de Valery LISKOVETS [Lis06]. Cette partie étend l'article [Pri15b]. Leurs formules mettent en bijection les  $2m^k$ -fonctions de parking et les automates acycliques (non-initiaux) sur un alphabet à  $k$  symboles. On construit une bijection explicite, généralisant encore une fois les travaux de D. FOATA et J. RIORDAN (§ 8.1, Théorème 8.1.12). Cette bijection est très expressive, et permet d'extraire nombre d'informations de l'automate immédiatement sur la  $2m^k$ -fonction de parking associée. En adaptant les techniques d'extraction de la première sous-partie, on obtient différents raffinements des automates acycliques (Lemme 8.2.2, § 8.3.1, § 8.3.2, § 8.4). Pour finir, on déduit, par double dénombrement, une formule d'énumération des automates acycliques minimaux, clôturant ainsi ce problème ouvert (Théorème 8.5.1).

## Réalisation polynomiale d'algèbre de Hopf

De nombreuses algèbres de Hopf combinatoires, FQSym, WQSym, PQSym, etc. se réalisent polynomialement. Les réalisations de ces algèbres se définissent toutes sensiblement de la même manière. On considère une application  $\phi$ , de l'ensemble mots dans une classe combinatoire, et on définit un sous-espace de l'algèbre libre étendu à partir de ses fibres. Dans la troisième et dernière partie (Partie III), on formalise cette théorie des réalisations polynomiales d'algèbres de Hopf à partir de telles applications, dites partitionnantes. Cette partie étend l'article en cours avec mon directeur de thèse Florent HIVERT [HP15].

On définit deux notions :

- le type d'alphabet : c'est une catégorie d'ensemble permettant de définir l'univers des alphabets utilisables pour réaliser une certaine algèbre de Hopf. Cette catégorie est enrichie d'un bifoncteur : la somme d'alphabet (Définition 10.1.1).
- l'application partitionnante : à partir d'un type d'alphabet, une application partitionnante définit une donnée qui associe, à chaque alphabet, une application de l'ensemble des mots dans une classe combinatoire fixé (Définition 10.2.2). Cette donnée satisfait quelques propriétés élémentaires sur la combinatoire des mots.

À partir de ces deux notions, on définit deux constructions d'algèbres de Hopf combinatoires en dualité (§ 10.3 et § 10.4).

On développe autour de ces applications différents opérateurs qui stabilisent la notion d'application partitionnante (§ 10.2.1), et on montre que les algèbres de Hopf construites sont automatiquement munies de certaines structures (comme les structures dendriformes ou le #-produit, Chapitre 11).

En guise d'application, on généralise la parkisation introduite par Jean-Christophe NOVELLI et Jean-Yves THIBON [NT04a]. Pour toute suite croissante  $\chi$ , on associe une application de parkisation qui associe, à tout mot, une  $\chi$ -fonction de parking (§ 12.2). On étudie dans quels cas cette application est une application partitionnante. On montre que l'application est partitionnante si et seulement si  $\chi : m \mapsto 1 + b(m - 1)$  (avec  $b \geq 0$

entier, Théorème [12.2.25](#)). Les uniques algèbres de Hopf obtenues correspondent ainsi à celles définies par les deux précédents auteurs dans [\[NT14\]](#).

# 1. Rappels et notations

## 1.1. Quelques classes combinatoires

Dans cette section, on rappelle quelques *classes combinatoires élémentaires* : les *partitions d'entiers*, les *partitions d'ensembles*, les *compositions d'entiers*, les *ordres linéaires* et aussi les *mots* et les *monoïdes libres* (sur des ensembles dénombrables). Les notations et définitions suivantes sont principalement extraites : de [FS09] et [BLL98].

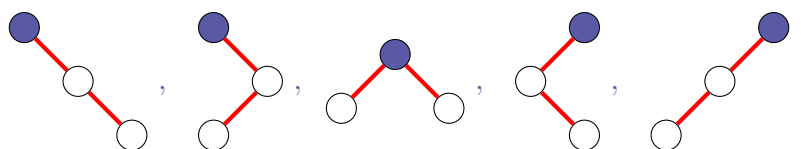
**Définition 1.1.1 :** Une *classe combinatoire* est un ensemble gradué  $\mathcal{C} = \bigsqcup_{n \geq 0} \mathcal{C}_n$  où chaque *composante graduée*  $\mathcal{C}_n$ , de degré  $n$ , est un ensemble fini. On note  $|\cdot| : \mathcal{C} \rightarrow \mathbb{N}$  l'application qui associe, à chaque élément  $s$  de  $\mathcal{C}$ , le degré  $n$  de la composante à laquelle il appartient,  $s \in \mathcal{C}_n$ .

À toute classe combinatoire  $\mathcal{C}$ , on associe une série formelle, appelée *série génératrice exponentielle* ou *ordinaire* :

$$\mathcal{C}(t) = \sum_{n \geq 0} \alpha_n \frac{t^n}{n!} \quad \text{ou} \quad \tilde{\mathcal{C}}(t) = \sum_{n \geq 0} \alpha_n t^n,$$

où  $\alpha_n := \#\mathcal{C}_n$  est le nombre de *structures combinatoires* dans la composante graduée  $\mathcal{C}_n$ . Nous verrons par la suite plus en détail les différentes utilisations de telles ou telles séries génératrices.

**Exemple 1.1.2 :** L'ensemble des arbres binaires est une *classe combinatoire*. Un arbre binaire est défini récursivement comme étant soit l'arbre vide soit un nœud, appelé *racine*, ayant deux *filles*, un droit et un gauche, qui sont respectivement des arbres binaires. L'application  $|\cdot|$  retourne le nombre de nœuds de l'arbre. Voici les cinq arbres binaires de la composante graduée de degré 3 (c'est-à-dire ayant 3 nœuds) :



où la racine est représentée colorée.

Cette classe combinatoire  $\tilde{\mathcal{B}}$  est définie récursivement. De cette définition récursive, on déduit (automatiquement) que la série génératrice des arbres binaires, notée  $\tilde{\mathcal{B}}(t)$ , satisfait l'équation :

$$\tilde{\mathcal{B}}(t) = 1 + t \tilde{\mathcal{B}}(t)^2.$$

En résolvant l'équation, on retrouve la série des nombres de Catalan :

$$\tilde{\mathcal{B}}(t) = \frac{1 - \sqrt{1 - 4t}}{2t} = 1 + t + 2t^2 + 5t^3 + 14t^4 + 42t^5 + 132t^6 + \dots \quad [\text{A000108}]$$

Soit  $f$  une *série formelle* (exponentielle ou ordinaire), on note  $[t^m]f(t)$  l'extraction du coefficient de degré  $m$  :

$$\text{si } f(t) = \sum_{n \geq 0} \alpha_n t^n \quad \text{alors } [t^m]f(t) = \alpha_m.$$

Dans la suite, on manipule de nombreux objets combinatoires définis comme des suites ou séquences (ordonnées ou non) de longueurs finies. Pour une telle structure  $s$ , on note  $\ell(s)$  la longueur de la suite/séquence sous-jacente.

**Remarque 1.1.3 :** Cette application  $\ell : \mathcal{C} \rightarrow \mathbb{N}$  est à distinguer de l'application  $|\cdot|$ . Par exemple, la partition  $\lambda := 4421$  est une partition de 11 définie par une séquence (non-ordonnée) de longueur 4 ( $|\lambda| = 11$  et  $\ell(\lambda) = 4$ ) (cf. prochaine sous-section, § 1.1.1).

### 1.1.1. Partitions d'entiers

Une *partition  $\lambda$  de l'entier  $n$*  est une suite finie décroissante d'entiers strictement positifs  $\lambda_1, \lambda_2, \dots, \lambda_k$  telle que :

$$|\lambda| := \lambda_1 + \lambda_2 + \dots + \lambda_k = n.$$

On note, par  $\lambda \vdash n$ , les partitions  $\lambda$  de l'entier  $n$ , la longueur de la suite qui définit  $\lambda$  est notée  $\ell(\lambda)$  et le nombre de parts de valeur  $k$  est noté  $\lambda^{(k)}$ .

**Exemple 1.1.4 :** Voici les premières partitions pour les entiers 0, 1, 2, 3, 4 et 5 :

$$\begin{aligned} \lambda \vdash 0 &= \{\epsilon\}, & \lambda \vdash 1 &= \{1\}, & \lambda \vdash 2 &= \{11, 2\}, & \lambda \vdash 3 &= \{111, 21, 3\}, \\ \lambda \vdash 4 &= \{1111, 211, 22, 31, 4\}, & \lambda \vdash 5 &= \{11111, 2111, 221, 311, 32, 41, 5\}, \end{aligned}$$

où  $\epsilon$  est la suite décroissante vide. La partition  $\lambda = 532221$  est de longueur  $6 = \ell(\lambda)$ . (**Notation :** par abus et/ou concision, on note  $\lambda_1 \lambda_2 \dots$  la partition  $\lambda$  définie par la séquence décroissante  $\lambda_1, \lambda_2, \dots$ )

### Série génératrice

La série génératrice des partitions d'entiers est :

$$\widetilde{\text{Par}}(t) = \prod_{n \geq 1} \frac{1}{1 - t^n} = 1 + t + 2t^2 + 3t^3 + 5t^4 + 7t^5 + 11t^6 + 15t^7 + 22t^8 + \dots \quad [\text{A000041}]$$

Pour donner une preuve de cette série, il suffit de remarquer que les partitions d'entiers sont en bijection avec les types d'isomorphismes de l'espèce des permutations (décrits dans § 1.5, Exemples 1.5.3 et 1.5.12), on déduit alors facilement la formule par spécialisation de la série indicatrice de cycles  $Z_{\mathcal{S}}$ .

### 1.1.2. Partitions d'ensembles

Soit  $U$  un ensemble fini. (On note  $S, T$  ou  $U$  les ensembles finis.)

Une *partition de l'ensemble*  $U$  est une famille de sous-ensembles non-vides disjoints de  $U$  telle que l'union est  $U$ . Sans ambiguïté, on note, par  $\lambda \vdash U$ , les partitions  $\lambda$  de l'ensemble  $U$  (de la même manière que les partitions d'entiers).

**Exemple 1.1.5 :** Voici les premières partitions pour les ensembles  $\emptyset, [1], [2], [3]$  et  $[4]$  (avec  $[n] = \{1, 2, \dots, n\}$ ) :

$$\begin{aligned} \lambda \vdash \emptyset &= \{ \emptyset \}, & \lambda \vdash [1] &= \{ \{1\} \}, & \lambda \vdash [2] &= \{ \{1 \mid 2\}, \{12\} \}, \\ \lambda \vdash [3] &= \{ \{1 \mid 2 \mid 3\}, \{12 \mid 3\}, \{23 \mid 1\}, \{13 \mid 2\}, \{123\} \}, \\ \lambda \vdash [4] &= \left\{ \begin{array}{l} \{1 \mid 2 \mid 3 \mid 4\}, \{12 \mid 3 \mid 4\}, \{13 \mid 2 \mid 4\}, \{14 \mid 2 \mid 3\}, \\ \{23 \mid 1 \mid 4\}, \{24 \mid 1 \mid 3\}, \{34 \mid 1 \mid 2\}, \{123 \mid 4\}, \\ \{124 \mid 3\}, \{134 \mid 2\}, \{234 \mid 1\}, \{1234\} \end{array} \right\}. \end{aligned}$$

(**Notation :** par abus et/ou concision, on décrit par  $\{\alpha\beta\cdots \mid \gamma\tau\cdots \mid \cdots\}$  la partition d'ensembles  $\{\{\alpha, \beta, \dots\}, \{\gamma, \tau, \dots\}, \dots\}$  où  $\mid$  sépare chaque sous-ensemble de la partition.)

#### Série génératrice

La série génératrice des partitions d'ensembles est :

$$\text{Par}(t) = \exp(e^t - 1) = 1 + t + 2\frac{t^2}{2!} + 5\frac{t^3}{3!} + 15\frac{t^4}{4!} + 52\frac{t^5}{5!} + 203\frac{t^6}{6!} + \cdots \quad [\text{A000110}]$$

Encore une fois en utilisant la définition en équation fonctionnelle de l'espèce des partitions (1.21), on obtient automatiquement la formule.

Les coefficients  $b_n$  de cette série sont les *nombres de Bell* :

$$b_n = \sum_{k=0}^n \left\{ \begin{matrix} n \\ k \end{matrix} \right\},$$

où  $\left\{ \begin{matrix} n \\ k \end{matrix} \right\}$  est le *nombre de Stirling* de seconde espèce :

$$\left\{ \begin{matrix} n \\ k \end{matrix} \right\} = \frac{1}{n!} \sum_{j=1}^k (-1)^{k-j} \binom{k}{j} j^n.$$

Ce *nombre de Stirling* compte le nombre de partitions en  $k$  parts distinctes d'un ensemble de cardinalité  $n$ . Cette dernière formule se démontre par inclusion-exclusion sur les surjections des ensembles de cardinalité  $n$  dans ceux de cardinalité  $k$ . Ce découpage des coefficients par les nombres de Stirling de seconde espèce raffine la série génératrice (univariée) en la suivante bivariée :

$$\exp(x(e^t - 1)) = \sum_{n,k} \left\{ \begin{matrix} n \\ k \end{matrix} \right\} x^k \frac{t^n}{n!}.$$

### 1.1.3. Compositions d'entiers

Une *composition*  $\pi$  de l'entier  $n$  est une suite (ordonnée) d'entiers strictement positifs  $\pi_1, \pi_2, \dots, \pi_k$  telle que :

$$|\pi| = \pi_1 + \pi_2 + \dots + \pi_k = n.$$

On note, par  $\pi \models n$ , les compositions  $\pi$  de l'entier  $n$  ; et pour toute composition  $\pi$ , on note  $\ell(\pi)$  la longueur de la suite associée. On définit  $\lambda(\pi)$  la surjection naturelle des compositions dans les partitions (où l'on identifie toutes les compositions  $\pi$  ayant le même (multi-)ensemble de  $\pi_i$ ).

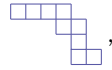
**Exemple 1.1.6 :** Voici les premières compositions des entiers 0, 1, 2, 3, 4 et 5 :

$$\begin{aligned} \pi \models 0 &= \{\epsilon\}, & \pi \models 1 &= \{1\}, & \pi \models 2 &= \{11, 2\}, & \pi \models 3 &= \{111, 21, 12, 3\}, \\ \pi \models 4 &= \{1111, 211, 121, 112, 22, 31, 13, 4\}, \\ \pi \models 5 &= \{11111, 2111, 1211, 1121, 1112, 221, 212, 122, 311, 131, 113, 32, 23, 41, 14, 5\}. \end{aligned}$$

On note aussi  $\pi(k)$  la somme partielle  $\pi_1 + \pi_2 + \dots + \pi_k$  des  $k$  premières parts de  $\pi$  avec comme conventions  $\pi(0) = 0$  et  $\pi(-1) = -1$ .

**Définition 1.1.7 :** Soit  $\pi = \pi_1 \pi_2 \dots \pi_k$  une composition de  $n$ . La *composition conjuguée* de  $\pi$ , noté  $\pi^\sim$ , est la composition obtenue par symétrie axiale sur l'axe  $y = x$  du ruban associé.

**Exemple 1.1.8 :** Le ruban de la composition 4212 est :



et sa conjuguée est la composition 132111 ayant pour ruban :



**Remarque 1.1.9 :** Dans la suite, on distinguera *composition* et *décomposition*. On appellera *décomposition* de  $n$  (de longueur  $k$ ) une suite (ordonnée) d'(e  $k$ ) entiers (0 inclus) de somme  $n$ . Par exemple 101204 est une décomposition de 8 (de longueur 6).

#### Ensemble de descentes et relation d'ordre

L'ensemble des sommes partielles  $\pi(k)$  définit l'*ensemble des descentes* de  $\pi$  :

$$D(\pi) = \{\pi(k) \mid 1 \leq k < \ell(\pi)\}.$$

**Exemple 1.1.10 :**

$$D(4234) = \{4, 6, 9\}, \quad D(21214) = \{2, 3, 5, 6\}.$$

**Remarque 1.1.11 :** La dernière somme partielle d'une composition  $\pi \models n$  vaut toujours  $n$ , elle n'est donc pas nécessaire pour définir (sans ambiguïté) l'ensemble des descentes d'une composition pour  $n$  fixé.

On dit que  $\tau$  est plus grossière que  $\pi$ , noté  $\pi \prec \tau$ , si et seulement si l'ensemble de descentes de  $\tau$  est inclu dans celui de  $\pi$  :

$$\pi \prec \tau \iff D(\pi) \supset D(\tau). \quad (1.1)$$

**Exemple 1.1.12 :** L'ordre suivant

$$4242 \prec 642 \prec 66 \prec (12)$$

se déduit de l'ordre d'inclusion :

$$\{4, 6, 10\} \supset \{6, 10\} \supset \{6\} \supset \{\}.$$

### Série génératrice

La série génératrice des compositions est :

$$\widetilde{\text{Bal}}(t) = \frac{1-t}{1-2t} = 1 + \sum_{n \geq 1} 2^{n-1} t^n = 1 + t + 2t^2 + 4t^3 + 8t^4 + 16t^5 + \dots \quad [\text{A011782}]$$

Par définition, une composition est une séquence d'entiers (non nuls). On peut exprimer l'ensemble des entiers comme la classe combinatoire où chaque composante graduée contient l'unique entier de même degré. Cette classe a pour série génératrice :

$$\mathbb{N}_{>0}(t) = \frac{1}{1-t} - 1 = \frac{t}{1-t}.$$

L'ensemble des compositions se traduit alors comme le monoïde libre (§ 1.1.6) engendré par l'étoile de Kleene (1.2) sur les entiers (non-nuls). D'après le formule (1.3) ci-après, on obtient la série,

$$\widetilde{\text{Bal}}(t) = \frac{1}{1 - \mathbb{N}_{>0}(t)} = \frac{1}{1 - \frac{t}{1-t}} = \frac{1-t}{1-2t}.$$

Dans la littérature, les compositions d'ensembles sont souvent appelées des *ballots*, **Bal**. De la même manière que les partitions d'entiers sont les types d'isomorphismes (cf. § 1.5) des partitions d'ensembles, les compositions d'entiers sont les types d'isomorphismes des compositions d'ensembles (cf. (1.22) d'où la notation  $\widetilde{\text{Bal}}(t)$  pour la série). Une manière simple de montrer que les coefficients sont  $2^{n-1}$  consiste à définir une composition comme un ruban et à les construire récursivement. On passe d'un ruban de taille  $n$  à un ruban de taille  $n + 1$  en ajoutant soit un pas horizontal soit un pas vertical.

### 1.1.4. Ordres linéaires

Les *ordres linéaires* d'un ensemble  $U$  sont tous les ordres totaux sur  $U$ . On note, par  $\ell \ll U$ , les ordres linéaires  $\ell$  de l'ensemble  $U$ .

**Exemple 1.1.13 :** Voici les premiers ordres linéaires pour les ensembles  $\emptyset$ ,  $[1]$ ,  $[2]$ ,  $[3]$  et  $[4]$  (notés comme des mots  $a_1 \cdots a_n$  signifiant  $a_1$  plus petit que  $a_2$ ,  $a_2$  plus petit que  $a_3$ , et, ainsi de suite jusqu'à  $a_n$ ) :

$$\begin{aligned} \ell \ll \emptyset &= \{\epsilon\}, \quad \ell \ll [1] = \{1\}, \quad \ell \ll [2] = \{12, 21\}, \\ \ell \ll [3] &= \{123, 132, 213, 231, 312, 321\}, \\ \ell \ll [4] &= \left\{ \begin{array}{l} 1234, 1243, 1324, 1342, 1423, 1432, 2134, 2143, \\ 2314, 2341, 2413, 2431, 3124, 3142, 3214, 3241, \\ 3412, 3421, 4123, 4132, 4213, 4231, 4312, 4321 \end{array} \right\} \end{aligned}$$

**Remarque 1.1.14 :** La longueur et le degré d'un ordre linéaire sont confondus ( $\ell(\ell) = |\ell|$ ). Les natures distinctes des deux objets : l'application  $\ell$  et la structure combinatoire  $\ell$  permettent de lever les ambiguïtés de notation possibles par la suite. La nature de ces objets est distincte.

### Série génératrice

La série génératrice des ordres linéaires est :

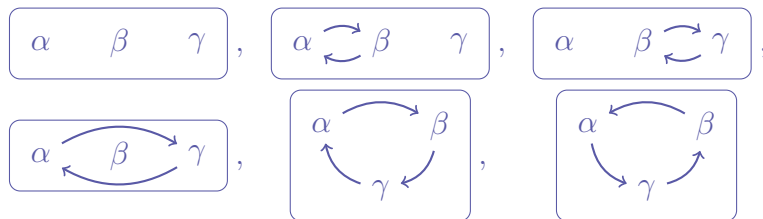
$$\mathcal{L}(t) = \frac{1}{1-t} = 1 + t + 2\frac{t^2}{2!} + 6\frac{t^3}{3!} + 24\frac{t^4}{4!} + 120\frac{t^5}{5!} + 720\frac{t^6}{6!} + \cdots \quad [\text{A000142}]$$

Le nombre d'ordres totaux possibles sur un ensemble de cardinalité  $n$  est évidemment  $n!$  :  $n$  choix du plus petit,  $n-1$  choix pour le suivant, etc.. Par ailleurs, les ordres linéaires sur  $U$  peuvent se définir récursivement comme un plus petit élément  $u$  suivi d'un ordre linéaire sur  $U \setminus \{u\}$  jusqu'à ce que l'ensemble  $U$  soit vide. On déduit alors la forme close  $(1-t)^{-1}$  en résolvant l'équation (cf. la définition récursive de l'espèce des ordres linéaires (1.17)).

### 1.1.5. Permutations et groupe symétrique

Une *permutation*  $\sigma$  de l'ensemble fini  $U$  est une endofonction de  $U$  bijective. On note  $\mathcal{S}[U]$  l'ensemble des permutations sur  $U$  ou simplement  $\mathcal{S}_n := \mathcal{S}[n]$  lorsque  $U = [n]$ .



**Exemple 1.1.15 :** Voici les six permutations de l'ensemble  $\{\alpha, \beta, \gamma\}$  (représentées par leur diagramme de transitions) :





Dans le cas où  $U = [n]$ , on confond la notation des ordres linéaires de  $\mathcal{L}[U]$  et celles des permutations  $\mathcal{S}[U]$  en identifiant  $\sigma$  à l'ordre linéaire  $\sigma(1)\sigma(2)\cdots\sigma(n)$ .

Chaque ensemble de permutations est naturellement muni d'une structure de groupe pour la composition : le *groupe symétrique*. L'identité est la permutation identité n'ayant que des points fixes, et, l'inverse est donné en inversant chaque transition dans le diagramme.

**Exemple 1.1.16 :** Dans l'exemple précédent, l'identité est  $\alpha \quad \beta \quad \gamma$  et les permutations  et  sont les uniques permutations inverses l'une de l'autre qui ne sont pas auto-inverses dans  $\mathcal{S}[U]$  pour  $U = \{\alpha, \beta, \gamma\}$ .

**Remarque 1.1.17 :** Les deux précédentes structures (ordre linéaire et permutation) sont usuellement confondues. Il est souvent agréable de noter une permutation par un ordre linéaire ou encore d'« inverser » un ordre linéaire. Ces conventions sont acceptables lorsque l'ensemble d'étiquettes est un ensemble d'entiers  $\{1, \dots, n\}$ . Dans le cas d'ensembles finis quelconques, la nature même de ces structures nécessite de les distinguer. Cette distinction des structures (de données) sera utile dans ce manuscrit.

### 1.1.6. Mots et monoïdes libres

Un *mot* sur  $\mathcal{A}$  est une séquence finie  $a_1a_2\cdots a_n$  d'éléments de  $U$ . On note  $\mathcal{A}^*$  l'ensemble engendré par l'étoile de Kleene sur  $\mathcal{A}$  :

$$\mathcal{A}^* = \sum_{n \geq 0} U^n = \{\epsilon\} \sqcup U \sqcup U^2 \sqcup U^3 \sqcup \dots \quad (1.2)$$

avec  $\epsilon$  le mot vide et  $\mathcal{A}^n$  est le produit cartésien de  $\mathcal{A}$  avec lui-même  $n$  fois :

$$\mathcal{A}^n = \mathcal{A} \times \mathcal{A}^{n-1} \text{ avec } \mathcal{A}^1 = U \text{ et } \mathcal{A}^0 = \{\epsilon\}.$$

On appelle *alphabet* et *lettres/symboles* respectivement l'ensemble  $\mathcal{A}$  et ses éléments. Un mot  $w$  de *longueur*  $n$  est une séquence de  $n$  lettres de l'alphabet ( $\ell(w) = n$ ).

On note  $\mathcal{A}(w)$  la restriction de l'ensemble  $\mathcal{A}$  à l'ensemble des symboles présents dans  $w$ .

**Exemple 1.1.18 :** Soit  $\mathcal{A} = \{a, b, c, d, e\}$ .

$$\mathcal{A}(abdda) = \{a, b, d\}.$$

### Série génératrice

Lorsque  $\mathcal{A}$  est une classe combinatoire, le monoïde libre  $\mathcal{A}^* = \bigsqcup_{n \geq 0} \mathcal{A}^n$  est aussi une classe combinatoire ayant pour série génératrice :

$$\mathcal{A}^*(t) = \frac{1}{1 - \mathcal{A}(t)} = \sum_{n \geq 0} \mathcal{A}(t)^n. \quad (1.3)$$

## Produit de concaténation

Le **produit de concaténation**  $\cdot$  associé sur  $\mathcal{A}^*$  est une opération associative homogène : pour tous mots  $u \in \mathcal{A}^m$  et  $v \in \mathcal{A}^n$ , la concaténation  $u \cdot v$ , simplement notée  $uv$ , est un mot de  $\mathcal{A}^{m+n}$  défini comme la séquence des éléments de  $u$  suivie par la séquence des éléments de  $v$ . L'ensemble  $\mathcal{A}^*$  muni de ce produit est appelé **monoïde libre**.

Cette opération permet de donner quelques nouvelles définitions :

- Un mot  $u = u_1 \cdots u_k$  est un **sous-mot** de  $w$  s'il existe un ensemble de  $k + 1$  mots  $x_1, x_2, \dots, x_{k+1}$  tels que  $w = x_1 \cdot u_1 \cdot x_2 \cdot u_2 \cdots x_k \cdot u_k \cdot x_{k+1}$ . Le mot  $x_1 \cdot x_2 \cdots x_{k+1}$  est le **sous-mot complémentaire** de  $u$  dans  $w$ .
- Un mot  $x$  est un **facteur** de  $w$  s'il existe deux mots  $u$  et  $v$  tels que  $w = u \cdot x \cdot v$ . Soit  $w = w_1 w_2 \cdots w_m$ , on note

$$w[i, j] := w_i w_{i+1} \cdots w_j \quad (1.4)$$

le facteur de  $w$  entre les positions  $i$  et  $j$ .

- Un mot  $x$  est un **préfixe** de  $w$  si  $x$  est un facteur de  $w$  tel que  $u = \epsilon$  ; de même  $x$  est un **suffixe** de  $w$  si  $x$  est un facteur de  $w$  tel que  $v = \epsilon$ .

## Produit de mélange

Le produit de concaténation est une opération ensembliste sur les mots. On sera aussi amené à considérer une seconde opération sur les combinaisons linéaires (formelles) de mots : le **produit de mélange**  $\sqcup$ . Cette opération linéaire consiste à associer à deux mots  $u \in \mathcal{A}^m$  et  $v \in \mathcal{A}^n$ , la somme de tous les mots  $w \in \mathcal{A}^{m+n}$  tels que  $u$  et  $v$  sont des **sous-mots complémentaires** de  $w$  (en autorisant  $w$  à apparaître autant de fois qu'il existe de découpage de  $u$  en  $m + 1 := |v| + 1$  facteurs  $x_1, \dots, x_{m+1}$  tel que  $w = x_1 \cdot v_1 \cdot x_2 \cdots x_m \cdot v_m \cdot x_{m+1}$ ). Autrement présenté, cette opération se définit récursivement comme une opération ayant le mot vide comme élément neutre et :

$$a \cdot u \sqcup b \cdot v = a \cdot (u \sqcup b \cdot v) + b \cdot (a \cdot u \sqcup v) \quad (1.5)$$

avec  $a, b \in \mathcal{A}$  et  $u, v \in \mathcal{A}^*$  (et où le produit de concaténation  $\cdot$  est étendu linéairement).

**Proposition 1.1.19 :** Le **produit de mélange** est associatif.

PREUVE : Le mot vide est l'élément neutre pour l'opération donc on montre l'associativité sur des mots non-vide. Soient  $a, b, c$  trois lettres. On a bien que  $(a \sqcup b) \sqcup c = a \sqcup (b \sqcup c)$ . Par induction sur la longueur des mots  $u, v, w$ , on a alors

$$\begin{aligned} (a \cdot u \sqcup b \cdot v) \sqcup c \cdot w &= a \cdot [(u \sqcup b \cdot v) \sqcup c \cdot w] + b \cdot [(a \cdot u \sqcup v) \sqcup c \cdot w] + \\ &\quad c \cdot [(a \cdot u \sqcup b \cdot v) \sqcup w] \\ &= a \cdot [u \sqcup (b \cdot v \sqcup c \cdot w)] + b \cdot [a \cdot u \sqcup (v \sqcup c \cdot w)] + \\ &\quad c \cdot [a \cdot u \sqcup (b \cdot v \sqcup w)] \\ &= a \cdot u \sqcup (b \cdot v \sqcup c \cdot w), \end{aligned}$$

pour toutes lettres  $a, b, c$  et tous mots  $u, v, w$ . ■

## Action du groupe symétrique

Soit  $\sigma : [n] \rightarrow [n]$  une bijection et soit  $w$  un mot de longueur  $n$ . Le groupe symétrique agit naturellement (à droite) sur les mots par :

$$w \cdot \sigma = w_{\sigma(1)} \cdot w_{\sigma(2)} \cdots w_{\sigma(n)}. \quad (1.6)$$

**Définition 1.1.20 :** On définit  $\text{Sh}_{m,n}$  comme l'ensemble des bijections  $\sigma$  de  $[m+n]$  telles que la restriction  $\sigma|_{[m]}$  de  $\sigma$  au domaine  $[m]$  et son complémentaire soient respectivement les bijections identités sur  $[m]$  et  $\{m+1, \dots, m+n\}$ .

Le *produit de mélange* se définit de manière équivalente à partir de  $\text{Sh}_{m,n}$  : soient  $u$  et  $v$  deux mots respectivement de longueurs  $m$  et  $n$ .

$$u \sqcup v = \sum_{\sigma \in \text{Sh}_{m,n}} uv \cdot \sigma. \quad (1.7)$$

### 1.1.7. Relations d'équivalences

Soit  $U$  un ensemble. Une *relation d'équivalence*  $\sim$  sur  $U$  est une relation binaire réflexive ( $x \sim x$ ), symétrique (si  $x \sim y$  alors  $y \sim x$ ) et transitive (si  $x \sim y$  et  $y \sim z$  alors  $x \sim z$ ).

#### Homogénéité

Soit  $\mathcal{C} := \bigoplus_{n \geq 0} \mathcal{C}_n$  une classe combinatoire ou plus généralement un ensemble gradué.

Une relation d'équivalence  $\sim$  sur  $\mathcal{C}$  est dite *homogène* si  $\sim$  est définie comme l'union disjointe des relations  $\sim_n$  sur  $\mathcal{C}_n$ , pour tout  $n$ .

#### Union

Soient  $\sim$  et  $\approx$  deux relations d'équivalences.

La *clôture transitive de l'union* des relations  $\sim$  et  $\approx$  est la relation  $\equiv := \sim \vee \approx$  définie par :  $u \equiv v$  si et seulement s'il existe une suite finie ( $u =: w_1, w_2, \dots, w_{m+1} := v$ ) telle que  $w_i \sim w_{i+1}$  ou  $w_i \approx w_{i+1}$ , pour tout  $i \in [m]$ .

#### Intersection

L'*intersection* des relations d'équivalence  $\sim$  et  $\approx$  est la relation  $\equiv := \sim \wedge \approx$  définie par

$$u \equiv v \quad \text{si et seulement si} \quad u \sim v \quad \text{et} \quad u \approx v.$$

## 1.2. Fonctions symétriques

Un(e) polynôme/fonction en  $n$  variables  $f(x_1, x_2, \dots, x_n)$  est *symétrique* s'il/elle est invariant(e) par permutations :

$$f(x_1, x_2, \dots, x_n) = f(x_{\sigma(1)}, x_{\sigma(2)}, \dots, x_{\sigma(n)}), \quad \text{pour toute bijection } \sigma \in \mathcal{S}_n.$$

Par exemple, le polynôme

$$x_1 x_2^2 x_3^3 + x_1 x_2^3 x_3^2 + x_1^2 x_2 x_3^3 + x_1^2 x_2^3 x_3 + x_1^3 x_2 x_3^2 + x_1^3 x_2^2 x_3 \quad (1.8)$$

est *symétrique*. Ces polynômes apparaissent dans de nombreux problèmes liés à la combinatoire, l'algèbre, la géométrie ou encore à la théorie des représentations du groupe symétrique. Leur importance fondamentale en combinatoire algébrique a donné naissance à l'algèbre (de Hopf) des *fonctions symétriques*, notée  $\text{Sym}$  ou encore  $\Lambda$ .

Nous donnons ici un rapide aperçu de quelques bases et propriétés de ces fonctions issu du premier chapitre de [Mac98]. Soit  $\mathbb{X}$  un ensemble infini de variables :

$$\mathbb{X} = x_1 + x_2 + x_3 + x_4 + \dots$$

L'algèbre des fonctions symétriques est donc l'algèbre des séries formelles de degrés bornés en  $\mathbb{X}$  invariantes pour l'action du groupe symétrique. La base la plus naturelle de cette algèbre est celle des *fonctions monomiales* :

### 1.2.1. Fonctions monomiales

Cette première base est obtenue en symétrisant les monômes. (Par exemple, le polynôme (1.8) est le symétrisé du monôme  $x_1^2 x_2 x_3^3$  sur un ensemble de variables restreint à  $x_1, x_2, x_3$ .) La base  $(m_\lambda)$  des *fonctions monomiales* est alors définie par :

$$m_\lambda(\mathbb{X}) = \sum_{\substack{i_1 < i_2 < \dots < i_k \\ \sigma}} x_{i_{\sigma(1)}}^{\lambda_1} x_{i_{\sigma(2)}}^{\lambda_2} \dots x_{i_{\sigma(k)}}^{\lambda_k} = x_1^{\lambda_1} x_2^{\lambda_2} \dots x_k^{\lambda_k} + x_2^{\lambda_1} x_1^{\lambda_2} \dots x_k^{\lambda_k} + \dots$$

avec  $k := \ell(\lambda)$  (le nombre de parts de  $\lambda$ ) et la somme parcourt toutes les permutations  $\sigma$  de  $\lambda$ .

### 1.2.2. Sommes de puissances

La base  $(p_\lambda)$  des *sommes de puissances* est définie par la série génératrice :

$$\psi_t(\mathbb{X}) = \sum_{n \geq 1} p_n(\mathbb{X}) t^{n-1} := \sum_{x \in \mathbb{X}} x(1 - xt)^{-1}$$

d'où

$$p_n(\mathbb{X}) = \sum_{x \in \mathbb{X}} x^n = x_1^n + x_2^n + x_3^n + \dots$$

### 1.2.3. Fonction élémentaires

La base  $(e_\lambda)$  des *fonctions élémentaires* est définie par la série génératrice :

$$\lambda_t(\mathbb{X}) = \sum_{n \geq 0} e_n(\mathbb{X}) t^n := \prod_{x \in \mathbb{X}} (1 + xt)$$

d'où

$$e_n(\mathbb{X}) = \sum_{i_1 < i_2 < \dots < i_n} x_{i_1} x_{i_2} \dots x_{i_n} = x_1 x_2 \dots x_n + x_1 x_3 \dots x_{n+1} + \dots$$

### 1.2.4. Fonctions homogènes

La base  $(h_\lambda)$  des *fonctions homogènes* est définie par la série génératrice :

$$\sigma_t(\mathbb{X}) = \sum_{n \geq 0} h_n(\mathbb{X}) t^n := \prod_{x \in \mathbb{X}} (1 - xt)^{-1} \quad (1.9)$$

d'où

$$h_n(\mathbb{X}) = \sum_{i_1 \leq i_2 \leq \dots \leq i_n} x_{i_1} x_{i_2} \dots x_{i_n} = x_1^n + x_2^n + \dots + x_1^2 x_3 \dots x_n + x_1 x_2 \dots x_n + \dots$$

Et quelque soit la base considérée parmi ces trois dernières, on définit :

$$f_\lambda := f_{\lambda_1} f_{\lambda_2} \dots f_{\lambda_k}, \quad \text{pour toute partition } \lambda = \lambda_1 \lambda_2 \dots \lambda_k,$$

où  $(f_\lambda)$  est la base  $(p_\lambda)$ ,  $(e_\lambda)$  ou  $(h_\lambda)$ .

### 1.2.5. Quelques égalités

On a les égalités :

$$\sum_{n \geq 0} h_n = \exp \left( \sum_{k \geq 1} \frac{p_k}{k} \right), \quad \sum_{n \geq 0} e_n = \exp \left( \sum_{k \geq 1} (-1)^{k-1} \frac{p_k}{k} \right)$$

qui donnent les changements de bases suivants :

$$h_n = \sum_{\lambda \vdash n} \frac{p_\lambda}{z_\lambda}, \quad e_n = \sum_{\lambda \vdash n} (-1)^{n-\ell(\lambda)} \frac{p_\lambda}{z_\lambda} \quad (1.10)$$

avec

$$z_\lambda := 1^{\lambda^{(1)}} 2^{\lambda^{(2)}} \dots k^{\lambda^{(k)}} \cdot \lambda^{(1)}! \lambda^{(2)}! \dots \lambda^{(k)}! . \quad (1.11)$$

On a aussi le changement de base suivant :

$$h_n = \sum_{\lambda \vdash n} m_\lambda .$$

## 1.2.6. $Sym_n$ et spécialisations

### Projection sur les polynômes en $n$ variables

On note  $Sym_n$  la projection des fonctions symétriques en  $n$  variables.

**Exemple 1.2.1 :** Le polynôme (1.8) est la projection de la monomiale  $m_{321}$  sur  $Sym_3$ .

### Spécialisation

On note  $m_{\lambda}(k)$  la *spécialisation de l'alphabet* où les variables  $x_1, x_2, \dots, x_k$  sont substituées par 1 et  $x_i$  par 0 pour tout  $i > k$  dans  $m_{\lambda}(\mathbb{X})$ . Autrement dit, cette spécialisation correspond à la projection  $Sym_k$  où toutes les variables sont substituées par 1. On alors :

$$m_{\lambda}(k) = \binom{\ell(\lambda)}{\lambda^{(1)}, \dots, \lambda^{(n)}} \binom{k}{\ell(\lambda)}. \quad (1.12)$$

pour toute partition  $\lambda \vdash n$ . Le multinomial correspond au nombre de manières distinctes de permuter un monôme et le binomial correspond au nombre de manière de choisir ces variables.

**Exemple 1.2.2 :**

$$m_{211}(5) = \underbrace{1^2 \times 1 \times 1 + 1 \times 1^2 \times 1 + 1 \times 1 \times 1^2}_{\binom{\ell(211)}{2,1}} + 1^2 \times 1 \times 1 + \dots = 30$$

$\underbrace{\hspace{15em}}_{\binom{5}{\ell(211)}}$

### $q$ -spécialisation

On note  $m_{\lambda}([k]_q)$  la  *$q$ -spécialisation de l'alphabet* où les variables  $x_i$  (avec  $1 \leq i \leq k$ ) sont substituées par  $q^{i-1}$  et par 0 dans les autres cas :

$$m_{\lambda}([k]_q) = \sum_{\substack{0 \leq i_1 < i_2 < \dots < i_m < k \\ \sigma \in \mathcal{S}_m}} \prod_{j=1}^m q^{\lambda_{\sigma(j)} i_j},$$

où  $\lambda = \lambda_1 \lambda_2 \dots \lambda_m$  et les  $i_j$  sont des entiers.

**Exemple 1.2.3 :**

$$\begin{aligned} m_{211}([5]_q) &= ((q^0)^2 q^1 q^2 + q^0 (q^1)^2 q^2 + q^0 q^1 (q^2)^2) + ((q^0)^2 q^1 q^3 + q^0 (q^1)^2 q^3 + \dots) + \dots \\ &= q^{13} + 2q^{12} + 4q^{11} + 2q^{10} + 4q^9 + 4q^8 + 4q^7 + 2q^6 + 4q^5 + 2q^4 + q^3 \end{aligned}$$

## 1.3. Fonctions symétriques non-commutatives

L'algèbre de Hopf des *fonctions symétriques non-commutatives*, notée **NCSF**, est une analogue de l'algèbre des fonctions symétriques indexée par les compositions d'entiers. À partir des travaux sur la théorie des quasi-déterminants de Israel M. GELFAND et Vladimir S. RETAKH [GR91], cette dernière a été introduite par I. M. GELFAND, Daniel KROB, Alain LASCoux, Bernard LECLERC, V. S. RETAKH et Jean-Yves THIBON dans [Gel+95] et largement étudiée dans le reste de l'« encyclopédie » de **NCSF** en sept volumes [KLT97 ; Duc+97 ; KT97 ; KT99 ; DHT02 ; Duc+11b]. Elle permet de relever, au niveau non-commutatif, un grand nombre d'applications classiques des fonctions symétriques.

Dans cette section, on introduit trois bases : celle des fonctions complètes ( $\mathbf{S}^\pi$ ), des fonctions rubans ( $\mathbf{R}_\pi$ ) et celle des fonctions élémentaires ( $\Lambda^\pi$ ) ; ainsi que les principaux opérateurs : le produit, le coproduit et le produit interne.

### 1.3.1. Bases

#### Fonctions élémentaires

La base des *fonctions élémentaires* ( $\Lambda^\pi$ ) se définit par la série génératrice :

$$\lambda_t(\mathbb{A}) = \sum_{n \geq 0} \Lambda^n(\mathbb{A}) t^n. \quad [\text{Gel+95, Éq. 21}]$$

#### Fonctions complètes

La base des *fonctions complètes* ( $\mathbf{S}^\pi$ ) se définit à partir de la série inverse de  $\lambda_{-t}$  :

$$\sigma_t(\mathbb{A}) = \sum_{n \geq 0} \mathbf{S}^n(\mathbb{A}) t^n = \lambda_{-t}(\mathbb{A})^{-1}. \quad [\text{Gel+95, Éq. 22}]$$

Dans ces deux bases ( $\Lambda^\pi$ ) et ( $\mathbf{S}^\pi$ ), on définit le produit par :

$$\Lambda^\pi = \Lambda^{\pi_1} \Lambda^{\pi_2} \dots \Lambda^{\pi_k} \quad \text{et} \quad \mathbf{S}^\pi = \mathbf{S}^{\pi_1} \mathbf{S}^{\pi_2} \dots \mathbf{S}^{\pi_k}, \quad [\text{Gel+95, Définition 3.4}]$$

pour toute composition  $\pi := \pi_1 \pi_2 \dots \pi_k$  ; et le coproduit par :

$$\Delta(\Lambda^n) = \sum_{i+j=n} \Lambda^i \otimes \Lambda^j \quad \text{et} \quad \Delta(\mathbf{S}^n) = \sum_{i+j=n} \mathbf{S}^i \otimes \mathbf{S}^j, \quad [\text{Gel+95, Proposition 3.8}]$$

pour tout entier  $n$ .

**Exemple 1.3.1 :**

$$\Lambda^{312} \Lambda^{54} = \Lambda^{31254} \quad \text{et} \quad \mathbf{S}^{98} \mathbf{S}^{271} = \mathbf{S}^{98271}$$

et

$$\begin{aligned} \Delta(\Lambda^{31}) &= 1 \otimes \Lambda^{31} + \Lambda^1 \otimes \Lambda^{21} + \Lambda^1 \otimes \Lambda^3 + \Lambda^{11} \otimes \Lambda^2 + \Lambda^2 \otimes \Lambda^{11} + \\ &\quad \Lambda^{21} \otimes \Lambda^1 + \Lambda^3 \otimes \Lambda^1 + \Lambda^{31} \otimes 1, \\ \Delta(\mathbf{S}^{111}) &= 1 \otimes \mathbf{S}^{111} + 3\mathbf{S}^1 \otimes \mathbf{S}^{11} + 3\mathbf{S}^{11} \otimes \mathbf{S}^1 + \mathbf{S}^{111} \otimes 1 \end{aligned}$$

Les changements de base entre ces deux bases se définissent par :

$$\mathbf{S}^\pi = \sum_{\tau \prec \pi} (-1)^{\ell(\tau) - |\pi|} \Lambda^\tau \quad \text{et} \quad \Lambda^\pi = \sum_{\tau \prec \pi} (-1)^{\ell(\tau) - |\pi|} \mathbf{S}^\tau, \quad [\text{Gel+95, Proposition 4.3}]$$

(cf. la définition de l'ordre  $\prec$  (1.1)).

**Exemple 1.3.2 :**

$$\begin{aligned} \mathbf{S}^{2121} &= \Lambda^{2121} - \Lambda^{21111} - \Lambda^{11121} + \Lambda^{111111}, \\ \Lambda^{31} &= \mathbf{S}^{1111} - \mathbf{S}^{121} - \mathbf{S}^{211} + \mathbf{S}^{31}. \end{aligned}$$

L'image commutative des fonctions élémentaires et complètes non-commutatives s'envoient respectivement sur les fonctions élémentaires et les fonctions complètes des fonctions symétriques.

### Fonctions de Schur rubans

Les *fonctions de Schur rubans* ( $\mathbf{R}_\pi$ ) sont définies à partir du changement de base :

$$\mathbf{R}_\pi = \sum_{\tau \prec \pi} (-1)^{\ell(\pi) + \ell(\tau)} \mathbf{S}^\tau \quad \text{et} \quad \mathbf{S}^\pi = \sum_{\pi \prec \tau} \mathbf{R}_\tau \quad [\text{Gel+95, Proposition 4.13}]$$

ou

$$\mathbf{R}_\pi = \sum_{\pi^\sim \prec \bar{\tau}} (-1)^{\ell(\pi^\sim) - \ell(\tau)} \Lambda^\tau \quad \text{et} \quad \Lambda^\pi = \sum_{\bar{\pi} \prec \tau^\sim} \mathbf{R}_\tau \quad [\text{Gel+95, Proposition 4.20}]$$

où  $\bar{\tau} := \tau_k \tau_{k-1} \cdots \tau_1$  est la composition miroir de  $\tau = \tau_1 \cdots \tau_{k-1} \tau_k$ .

**Exemple 1.3.3 :**

$$\begin{aligned} \mathbf{R}_{322} &= \mathbf{S}^{322} - \mathbf{S}^{34} - \mathbf{S}^{52} + \mathbf{S}^7, \\ &= \mathbf{R}_{12211^\sim} = \Lambda^{11221} - \Lambda^{1123} - \Lambda^{1141} + \Lambda^{115} - \Lambda^{1321} + \Lambda^{133} + \Lambda^{151} - \Lambda^{16} - \Lambda^{2221} + \\ &\quad \Lambda^{223} + \Lambda^{241} - \Lambda^{25} + \Lambda^{421} - \Lambda^{43} - \Lambda^{61} + \Lambda^7 \end{aligned}$$

Le produit de ces éléments est défini par :

$$\mathbf{R}_\pi \mathbf{R}_\tau = \mathbf{R}_{\pi \triangleright \tau} + \mathbf{R}_{\pi \cdot \tau}, \quad [\text{Gel+95, Proposition 3.13}]$$

où  $\pi \triangleright \tau := \pi_1 \cdots \pi_{j-1} (\pi_j + \tau_1) \tau_2 \cdots \tau_k$  et  $\pi \cdot \tau := \pi_1 \cdots \pi_j \tau_1 \cdots \tau_k$ , pour toutes compositions  $\pi := \pi_1 \cdots \pi_j$  et  $\tau := \tau_1 \cdots \tau_k$ .

**Exemple 1.3.4 :**

$$\mathbf{R}_{24} \mathbf{R}_{1413} = \mathbf{R}_{25413} + \mathbf{R}_{241413}.$$



### 1.3.2. Produit interne

Le *produit interne* des fonctions symétriques non-commutatives correspond aux produits des algèbres de descentes de Louis SOLOMON [Sol76]. Une manière élégante d'exprimer ce produit consiste à faire intervenir des matrices d'entiers :

Soient  $\pi$  et  $\tau$  respectivement deux compositions ayant respectivement  $m$  et  $n$  parts. On note  $M(\pi, \tau)$  l'ensemble des matrices  $(\mathbf{m}_{i,j})$  d'ordre  $(m, n)$  telles que

$$\sum_{i=1}^m \mathbf{m}_{i,j} = \tau_j \quad \text{et} \quad \sum_{j=1}^n \mathbf{m}_{i,j} = \pi_i$$

pour respectivement tout  $i \in [n]$  et tout  $j \in [m]$ .

**Exemple 1.3.5 :**

$$M(32, 122) = \left\{ \begin{pmatrix} 1 & 2 & 0 \\ 0 & 0 & 2 \end{pmatrix}, \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 2 \\ 0 & 2 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 2 & 1 \\ 1 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 0 & 1 & 2 \\ 1 & 1 & 0 \end{pmatrix} \right\}.$$

À chaque matrice  $(\mathbf{m}_{i,j})$ , une lecture ligne par ligne (de gauche à droite) des coefficients non-nuls définit une nouvelle composition,  $\pi(\mathbf{m}_{i,j})$ .

**Exemple 1.3.6 :**

$$\begin{aligned} \pi \begin{pmatrix} 1 & 2 & 0 \\ 0 & 0 & 2 \end{pmatrix} &= 122, & \pi \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \end{pmatrix} &= 11111, & \pi \begin{pmatrix} 1 & 0 & 2 \\ 0 & 2 & 0 \end{pmatrix} &= 122, \\ \pi \begin{pmatrix} 0 & 2 & 1 \\ 1 & 0 & 1 \end{pmatrix} &= 2111, & \pi \begin{pmatrix} 0 & 1 & 2 \\ 1 & 1 & 0 \end{pmatrix} &= 1211. \end{aligned}$$

Le produit interne se définit alors par :

$$\mathbf{S}^\pi \diamond \mathbf{S}^\tau = \sum_{(\mathbf{m}_{i,j}) \in M(\pi, \tau)} \mathbf{S}^{\pi(\mathbf{m}_{i,j})}. \quad [\text{Gel+95, Proposition 5.1}]$$

**Remarque 1.3.7 :** Ces matrices sont appelées *matrices tassées d'entiers*. Cette description du produit interne dans les fonctions symétriques non-commutatives provient du produit interne d'une algèbre de Hopf plus grosse sur ces matrices tassées [Duc+11a, § 5].

**Exemple 1.3.8 :**

$$\mathbf{S}^{32} \diamond \mathbf{S}^{122} = 2\mathbf{S}^{122} + \mathbf{S}^{11111} + \mathbf{S}^{2111} + \mathbf{S}^{1211}.$$

La série  $\sigma_1(\mathbb{A})$  est l'élément neutre pour le produit interne :

$$F \diamond \sigma_1(\mathbb{A}) = F = \sigma_1(\mathbb{A}) \diamond F.$$

### 1.3.3. Opérateurs d'Adams

Les puissances du produit de convolution  $\text{Id}^{*k}$  de l'identité (cf. (9.6), § 9.2.1) d'une bialgèbre sont appelées les *opérateurs d'Adams* [KLT97 ; AL15] :

$$\text{Id}^{*k} = \mu^k \circ \Delta^k, \quad \text{où} \quad \begin{cases} \Delta^k := \begin{cases} \Delta & \text{si } k = 1, \\ (\Delta^{k-1} \otimes \text{Id}) \circ \Delta & \text{sinon ;} \end{cases} \\ \mu^j := \begin{cases} \mu & \text{si } k = 1, \\ \mu^{k-1} \circ (\mu \otimes \text{Id}^{k-1}) & \text{sinon,} \end{cases} \end{cases}$$

avec  $\mu$  l'opérateur de produit.

**Exemple 1.3.9 :**

$$\begin{aligned} \text{Id}^{1*}(\mathbf{S}^3) &= \mu(\mathbf{S}^3 \otimes 1 + \mathbf{S}^2 \otimes \mathbf{S}^1 + \mathbf{S}^1 \otimes \mathbf{S}^2 + 1 \otimes \mathbf{S}^3), \\ &= 2\mathbf{S}^3 + \mathbf{S}^{21} + \mathbf{S}^{12}; \end{aligned}$$

$$\begin{aligned} \text{Id}^{2*}(\mathbf{S}^2) &= (\mu^2 \circ \Delta)(\mathbf{S}^2 \otimes 1 + \mathbf{S}^1 \otimes \mathbf{S}^1 + 1 \otimes \mathbf{S}^2) \\ &= \mu^2(\mathbf{S}^2 \otimes 1 \otimes 1 + \mathbf{S}^1 \otimes \mathbf{S}^1 \otimes 1 + 1 \otimes \mathbf{S}^2 \otimes 1 + \mathbf{S}^1 \otimes 1 \otimes \mathbf{S}^1 + \\ &\quad 1 \otimes \mathbf{S}^1 \otimes \mathbf{S}^1 + 1 \otimes \mathbf{S}^2 \otimes 1 + 1 \otimes \mathbf{S}^1 \otimes \mathbf{S}^1 + 1 \otimes 1 \otimes \mathbf{S}^2) \\ &= 4\mathbf{S}^2 + 4\mathbf{S}^{11} \end{aligned}$$

$$\text{Id}^{2*}(\mathbf{S}^4) = \mathbf{S}^{112} + \mathbf{S}^{121} + 3\mathbf{S}^{13} + \mathbf{S}^{211} + 3\mathbf{S}^{22} + 3\mathbf{S}^{31} + 3\mathbf{S}^4$$

**Proposition 1.3.10 :**

$$\text{Id}^{k*}(\mathbf{S}^n) = \sum_{\pi \models n} \binom{k+1}{\ell(\pi)} \mathbf{S}^\pi.$$

PREUVE : Par définition du coproduit de la fonction complète  $\mathbf{S}^n$ , le coefficient de la composition  $\pi$  correspond aux nombres de tenseurs de la forme

$$\mathbf{S}^{j_1} \otimes \mathbf{S}^{j_2} \otimes \dots \otimes \mathbf{S}^{j_{k+1}}$$

où la suite des  $j_i \neq 0$  est  $\pi$ . Ce nombre correspond au nombre de manière de choisir  $\ell(\pi)$  points parmi  $k+1$ . ■

### 1.3.4. Base monomiale des fonctions quasi-symétriques

L'algèbre de Hopf des fonctions symétriques non-commutatives est en dualité (graduée) avec l'algèbre de Hopf des *fonctions quasi-symétriques*, notée **QSym**, introduite par Ira M. GESSEL [Ges84].

Cette algèbre de Hopf, tout comme les fonctions symétriques, peut s'exprimer en termes de séries sur un nombre infini de variables  $\mathbb{X}$ . À la différence des fonctions symétriques, on considère  $\mathbb{X}$  comme un alphabet de variables totalement ordonnées.

La base en dualité des fonctions symétriques non-commutatives complètes ( $\mathbf{S}^\pi$ ) est la base des quasi-monomiales ( $\mathbf{M}_\pi$ ) de  $\mathbf{QSym}$  défini par :

$$\mathbf{M}_\pi(\mathbb{X}) := \sum_{i_1 < i_2 < \dots < i_k} x_{i_1}^{\pi_1} x_{i_2}^{\pi_2} \dots x_{i_k}^{\pi_k}. \quad (1.13)$$

**Exemple 1.3.11 :** Soit  $\mathbb{X} := x_1 < x_2 < x_3 < \dots$ .

$$\mathbf{M}_{121}(\mathbb{X}) = x_1^1 x_2^2 x_3^1 + x_1^1 x_2^2 x_4^1 + x_1^1 x_2^2 x_5^1 + \dots + x_1^1 x_3^2 x_4^1 + x_1^1 x_3^2 x_5^1 + \dots$$

Le produit de ces monomiales se décrit comme un produit de quasi-mélange des compositions. Il est défini par :

$$\pi_1 \pi \cup \tau_1 \tau := \pi_1(\pi \cup \tau_1 \tau) + \tau_1(\pi_1 \pi \cup \tau) + (\pi_1 + \tau_1)(\pi \cup \tau),$$

**Exemple 1.3.12 :**

$$\begin{aligned} \mathbf{M}_{121} \mathbf{M}_{21} = & 2\mathbf{M}_{12121} + 4\mathbf{M}_{12211} + 2\mathbf{M}_{1222} + 2\mathbf{M}_{1231} + 2\mathbf{M}_{1411} + \mathbf{M}_{142} + \\ & 2\mathbf{M}_{21121} + 2\mathbf{M}_{21211} + \mathbf{M}_{2122} + \mathbf{M}_{2131} + \mathbf{M}_{2221} + \mathbf{M}_{3121} + \\ & 2\mathbf{M}_{3211} + \mathbf{M}_{322} + \mathbf{M}_{331} \end{aligned}$$

Similairement aux spécialisations des fonctions symétriques monomiales (§ 1.2.6), on note  $\mathbf{M}_\pi(k)$  la *spécialisation d'alphabet* où les variables  $x_i$  de 1 à  $k$  sont substituées par 1 et les autres par 0. On a alors :

$$\mathbf{M}_\pi(k) = \binom{k}{\ell(\pi)}.$$

Et de même, on note  $\mathbf{M}_\pi([k]_q)$  la spécialisation où les variables  $x_i$  sont substituées en  $q^{i-1}$  pour tout  $1 \leq i \leq k$  et par 0 sinon.

### 1.3.5. Transformation d'alphabet

Dans [KLT97], Daniel KROB, Bernard LECLERC et Jean-Yves THIBON étudient les analogues des transformations d'alphabets des fonctions symétriques dans les fonctions symétriques non-commutatives. Ils utilisent ces transformations pour décrire les idempotents et les nilpotents dans les algèbres de descentes. Dans cette sous-partie, nous nous intéressons uniquement à la transformation de produit d'alphabets qui définit un relèvement non-commutatif du produit de Cauchy.

Soit  $\mathbb{X}$  un alphabet totalement ordonné (commutatif) et soit  $\mathbb{A}$  un second alphabet (virtuel) totalement ordonné (non-commutatif). On définit

$$\sigma_t(\mathbb{X} \cdot \mathbb{A}) = \sum_{n \geq 0} \mathbf{S}^n(\mathbb{X} \cdot \mathbb{A}) t^n := \prod_{x \in \mathbb{X}}^{\rightarrow} \sigma_{xt}(\mathbb{A}).$$

En développant le produit, on déduit

$$\mathbf{S}^n(\mathbb{X} \cdot \mathbb{A}) = \sum_{\substack{\pi \models n \\ \pi = \pi_1 \cdots \pi_k}} \sum_{j_1 < j_2 < \cdots < j_k} x_{j_1}^{\pi_1} x_{j_2}^{\pi_2} \cdots x_{j_k}^{\pi_k} \mathbf{S}^\pi(\mathbb{A}),$$

à partir de la définition des quasi-monomiales (1.13), on a l'identité :

$$\mathbf{S}^n(\mathbb{X} \cdot \mathbb{A}) = \sum_{\pi \models n} \mathbf{M}_\pi(\mathbb{X}) \mathbf{S}^\pi(\mathbb{A}).$$

En utilisant les spécialisations de la précédente sous-section, on déduit

**Proposition 1.3.13 :**

$$\mathbf{S}^n(k \cdot \mathbb{A}) = \text{Id}^{*k-1} \mathbf{S}^n = \sum_{\pi \models n} \binom{k}{\ell(\pi)} \mathbf{S}^\pi.$$

Par la suite, on utilise cette notation  $\mathbf{S}^n(k \cdot \mathbb{A})$  pour signifier le  $k - 1^{\text{ième}}$  opérateur d'Adams appliqué à  $\mathbf{S}^n$ . Cette notation est cohérente avec  $h_n(k \cdot \mathbb{X})$  dans les fonctions symétriques qui correspond aussi à l'application du  $k - 1^{\text{ième}}$  opérateur d'Adams à  $h_n$ .

Avec analogie avec les fonctions symétriques, on étend la transformation  $\mathbf{S}^n(k \cdot \mathbb{A})$  au  $q$ -entier :

$$\mathbf{S}^n([k]_q \cdot \mathbb{A}) = \sum_{\pi \models n} \mathbf{M}_\pi([k]_q) \mathbf{S}^\pi(\mathbb{A}).$$

## 1.4. Théorème de dénombrement de Redfield-Pólya

Dans cette section, on rappelle les outils d'énumérations d'ensembles finis  $U$  sous l'action d'un groupe (fini)  $G$  [Bur11 ; Red27 ; Pól37]. Pour se faire, on utilise quelques notions de théorie des groupes [Per82].

### 1.4.1. Notions de théorie des groupes

**Définition 1.4.1 :** Un **groupe**  $G$  est un ensemble muni d'une loi de composition interne associative,  $\cdot : G \times G \rightarrow G$ , pour laquelle il existe un élément neutre (noté  $1$ ) et un inverse.

Soit  $U$  un ensemble. On dit que le groupe  $G$  **opère** sur  $U$  si on s'est fixé une application :

$$\begin{aligned} G \times U &\longrightarrow U \\ (g, u) &\longmapsto g \cdot u, \end{aligned}$$

vérifiant

- pour tous  $g, g' \in G$  et tout  $u \in U$ , on a  $(g \cdot g') \cdot u = g \cdot (g' \cdot u)$ ,
- pour tout  $u \in U$ ,  $1 \cdot u = u$ .

On parle alors de l'action de  $G$  sur l'ensemble  $U$ . Considérer l'action un groupe  $G$  opérant sur un ensemble  $U$  est équivalent à considérer un morphisme de  $G$  dans le groupe des bijections de  $U$ , noté  $G(U)$ . Ce point va notamment apparaître dans la définition du *transport de structures* d'une espèce (§ 1.5 ou [BLL98]).

À tout groupe  $G$  opérant sur un ensemble  $U$ , on peut associer une relation d'équivalence  $R$  défini par :

$$uRu' \quad \text{si et seulement si} \quad \exists g \in G, \quad u = g \cdot u'.$$

On appelle *orbite* de  $u \in U$ , notée  $\omega(u)$ , la classe d'équivalence de  $u$  pour la relation  $R$  (réflexive :  $u = Id \cdot u$ , symétrique :  $u = g \cdot u'$ , d'où  $g^{-1} \cdot u = g^{-1} \cdot (g \cdot u) = (g^{-1} \cdot g) \cdot u = u$ , et, transitive :  $u = g \cdot v = g \cdot (g' \cdot w) = (g \cdot g') \cdot w$ ).

**Définition 1.4.2 :** Soit  $u \in U$ . Le *stabilisateur* de  $u$  est le sous-groupe  $S_u$  de  $G$  :

$$S_u = \{ g \in G \mid g \cdot u = u \}.$$

**Proposition 1.4.3 :** Lorsque  $G$  est fini, on a la relation suivante :

$$\#G = \#S_u \times \#\omega(u).$$

Cette proposition découle du théorème de Lagrange :

**Théorème 1.4.4 (Lagrange) :** Soit  $G$  un groupe fini. Pour tout sous-groupe  $H$  de  $G$ , le cardinal de  $H$  divise le cardinal de  $G$ .

**Définition 1.4.5 :** L'ensemble des points fixes de  $U$  par l'action de  $g \in G$  est l'ensemble :

$$\text{Fix } g = \{ u \in U \mid g \cdot u = u \}. \quad (1.14)$$

On note  $\text{fix } g := \#\text{Fix } g$  le nombre d'éléments de  $\text{Fix } g$ .

## 1.4.2. Lemme de Burnside

La *formule de Burnside* [Bur11] donne une première technique d'énumération du nombre d'orbites sous l'action d'un groupe fini  $G$  sur un ensemble fini  $U$ .

**Lemme 1.4.6 (Formule de Burside) :** Le nombre d'orbites de  $U$  sous l'action de  $G$  est donnée par la formule :

$$\frac{1}{\#G} \sum_{g \in G} \text{fix } g.$$

Pour montrer ce lemme, on utilise une propriété intermédiaire :

**Proposition 1.4.7 :** Soient  $u$  et  $v$  deux éléments de  $U$  d'une même orbite. Les sous-groupes stabilisateurs  $S_u$  et  $S_v$  ont même cardinalité :

$$\#S_u = \#S_v.$$

PREUVE : Soit  $h \in S_u$  et soit  $g \in G$  tel que  $g \cdot u = v$ . On montre alors  $g \cdot h \cdot g^{-1} \in S_v$  :

$$\begin{aligned} g \cdot (h \cdot (g^{-1} \cdot v)) &= g \cdot (h \cdot u) \\ &= (g \cdot h \cdot g^{-1}) \cdot v = g \cdot u = v. \end{aligned}$$

D'où  $\#S_u \geq \#S_v$ , et réciproquement, donc on a bien  $\#S_u = \#S_v$ . ■

On montre alors le lemme en utilisant la technique de double dénombrement :

PREUVE (LEMME 1.4.6) : Soit  $A$  l'ensemble :

$$A := \{ (g, u) \in G \times U \mid g \cdot u = u \}.$$

On peut décomposer cet ensemble de deux manières distinctes :

$$A = \bigsqcup_{u \in U} S_u \times \{u\} = \bigsqcup_{g \in G} \{g\} \times \text{Fix } g,$$

avec  $S_u$  le stabilisateur de  $u$ . Les unions disjointes permettent de transposer ces égalités aux cardinaux des ensembles, on a donc :

$$\#A = \sum_{u \in U} \#S_u = \sum_{g \in G} \text{fix } g.$$

On pose  $O$  l'ensemble des orbites de  $U$  sous l'action de  $G$ .

$$\begin{aligned} \#A &= \sum_{\omega(u) \in O} \sum_{u' \in \omega(u)} \#S_{u'}, \\ &= \sum_{\omega(u) \in O} \#\omega(u) \#S_u, && \text{(Proposition 1.4.7)} \\ &= \sum_{\omega(u) \in O} \#G, && \text{(Proposition 1.4.3)} \\ &= \#O \times \#G. \end{aligned}$$

D'où

$$\#O = \frac{1}{\#G} \sum_{g \in G} \text{fix } g. \quad \blacksquare$$

### 1.4.3. Théorème de Redfield–Pólya

Le théorème de dénombrement de John Howard REDFIELD [Red27], redécouvert indépendamment quelques années plus tard par George PÓLYA [Pól37], est une généralisation du Lemme 1.4.6 de BURNSIDE sur le nombre d'orbites de  $U$  sous l'action de  $G$ . Ce théorème a été popularisé pour ces applications dans de nombreux domaines : chimie, physique, etc. et en particulier en combinatoire énumérative. En combinatoire, cette

formule d'énumération est au centre de la théorie des espèces développée par André JOYAL.

La formule de REDFIELD-PÓLYA enrichit simplement d'une information supplémentaire la formule de William BURNSIDE : l'*indicateur de cycles*. Cet indicateur est un polynôme en plusieurs variables qui transporte l'information de l'action du groupe  $G$  sur l'ensemble  $U$  :

$$Z_U(a_1, a_2, \dots) = \sum_{g \in G} \text{fix } g \, a_1^{j_1(g)} a_2^{j_2(g)} \dots a_n^{j_n(g)}$$

où  $a_1, a_2, \dots, a_n$  sont des variables et les  $j_k(g)$  correspondent au nombre de  $k$ -cycles de la permutation associée à  $g$ .

En particulier, on peut facilement extraire de ce polynôme le nombre d'orbites en substituant chaque variable par 1. Cette substitution permet de retrouver exactement la formule de BURNSIDE. On peut aussi retrouver le nombre d'éléments de  $U$  en extrayant le coefficient associé au monôme  $a_1^n$ . En effet, ce monôme correspond à l'indicateur de cycle de la permutation identité qui fixe l'ensemble des éléments de  $U$ .

Dans la prochaine section, on utilise cet indicateur de cycles dans le cadre de la théorie des espèces.

## 1.5. Théorie des espèces

Dans toutes les branches des mathématiques ainsi qu'en informatique théorique, le concept de structures discrètes (ou structures de données) est fondamental. La *théorie des espèces*, introduite par André JOYAL (1981), a pour objectif de donner une description élégante de ces structures (de données) discrètes. Elle permet aussi de traiter de façon naturelle l'*action du groupe symétrique* sur ces structures.

Cette théorie fournit un cadre puissant pour énumérer aussi bien les structures combinatoires étiquetées (ex. partitions d'ensembles, compositions d'ensembles, etc.), que les structures non-étiquetées (ex. partitions d'entiers, compositions d'entiers, etc.) en enrichissant la notion de série génératrice : les *séries indicatrices de cycles*. Ces séries sont des séries dans les *fonctions symétriques* définies à partir du *théorème de dénombrement* de REDFIELD-PÓLYA [Red27 ; Pól37]. Un autre atout des espèces est qu'elles concilient les définitions constructives et récursives des structures avec la théorie de PÓLYA. Elle intègre un grand nombre d'opérations et constructions combinatoires usuelles (*union disjointe*, *produit cartésien*, *pointage*, etc.).

Les sous-sections suivantes rappellent les principales définitions et notations présentées dans [BLL98]. Nous utiliserons autant que possible le formalisme présenté dans ce livre.

### 1.5.1. Définition

Cette section reprend les définitions principales du premier chapitre de [BLL98]. Dans le langage de la théorie des catégories, une espèce (ensembliste) se définit simple-

ment comme un foncteur de la catégorie des ensembles finis avec les bijections vers la catégorie des ensembles finis. Sans utiliser ce formalisme, une espèce de structures (ensembliste) se définit de la manière suivante :

**Définition 1.5.1 :** Une *espèce (de structures)* est une donnée  $\mathbf{F}$  qui produit :

1. pour chaque ensemble fini  $U$ , un ensemble fini  $\mathbf{F}[U]$ ,
2. pour chaque bijection  $\sigma : U \rightarrow V$ , une fonction  $\mathbf{F}[\sigma] : \mathbf{F}[U] \rightarrow \mathbf{F}[V]$ .

telle que les propriétés fonctorielles suivantes soient satisfaites :

1. pour toutes bijections  $\sigma : U \rightarrow V$  et  $\tau : V \rightarrow W$ ,

$$\mathbf{F}[\tau \circ \sigma] = \mathbf{F}[\tau] \circ \mathbf{F}[\sigma],$$

2. à chaque fonction identité  $\text{Id}_U : U \rightarrow U$ ,

$$\mathbf{F}[\text{Id}_U] = \text{Id}_{\mathbf{F}[U]}.$$

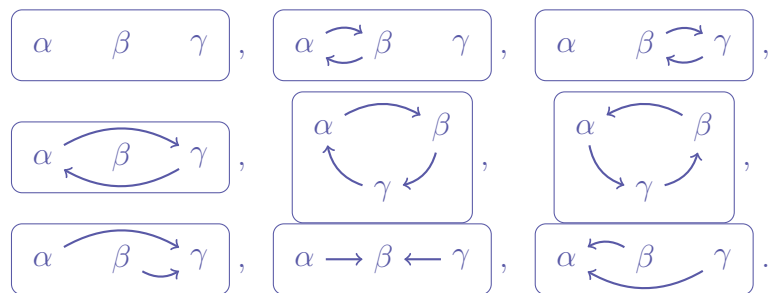
**Remarque 1.5.2 :**  $\mathbf{F}[\sigma]$  est une bijection, pour toutes espèce  $\mathbf{F}$  et bijection  $\sigma$ .

On appelle  $\mathbf{F}$ -*structure* (ou *structure de l'espèce  $\mathbf{F}$* ) sur  $U$  les éléments  $s \in \mathbf{F}[U]$ . La fonction  $\mathbf{F}[\sigma]$  est appelée le *transport* de  $\mathbf{F}$ -structures le long de  $\sigma$ . On notera souvent  $\sigma \cdot s$  pour désigner  $\mathbf{F}[\sigma](s)$ . Les  $\mathbf{F}$ -structures sont généralement appelées les *objets étiquetés* d'une « classe combinatoire » dans [FS09, Chapitre II]. Par la suite, on manipulera souvent les structures sur les ensembles finis d'entiers  $[n] = \{1, \dots, n\}$ , on notera alors  $\mathbf{F}[n]$  au lieu de  $\mathbf{F}[[n]]$  ou  $\mathbf{F}[\{1, \dots, n\}]$ .

**Exemple 1.5.3 :** Soit  $\text{End}$  l'espèce des endofonctions définit par :

- $\text{End}[U] := \{f \mid f : U \rightarrow U\}$ , pour tout ensemble fini  $U$ ,
- $\text{End}[\sigma] : f \mapsto \sigma \circ f \circ \sigma^{-1}$ , pour toute bijection  $\sigma : U \rightarrow V$ .

Voici l'ensemble des neuf endofonctions sur l'ensemble  $\{\alpha, \beta, \gamma\}$  :



Soit  $\mathcal{S}$  l'espèce des permutations définit comme  $\text{End}$  restreint aux bijections :

- $\mathcal{S}[U] := \{\sigma \mid \sigma : U \rightarrow U \text{ et } \sigma \text{ bijection}\}$  pour tout ensemble fini  $U$ ,
- $\mathcal{S}[\sigma] = \text{End}[\sigma]$ , pour toute bijection  $\sigma : U \rightarrow V$ .

Dans la suite, on manipulera fréquemment l'ensemble des permutations de l'ensemble  $[n] = \{1, \dots, n\}$ , on notera simplement  $\mathcal{S}_n$  à la place de  $\mathcal{S}[n]$  (cf. Exemple 1.1.15).

La notion d'action du groupe symétrique agissant sur les structures est intégrée dans la définition d'une espèce.



**Définition 1.5.4 :** Soient  $s \in \mathbf{F}[U]$  et  $t \in \mathbf{F}[V]$  deux  $\mathbf{F}$ -structures. Une bijection  $\sigma : U \rightarrow V$  est un **isomorphisme** de  $s$  à  $t$  si  $\mathbf{F}[\sigma](s) = \sigma \cdot s = t$ . On dit alors que  $s$  et  $t$  ont même **type d'isomorphisme**. Un isomorphisme qui fixe  $s$  est un **automorphisme** de  $s$ .

Dans le cas où  $U = V$ ,  $s$  et  $t$  ont même type d'isomorphisme s'ils appartiennent à la même orbite sous l'action du groupe symétrique (cf. § 1.4.1). On dénote  $\sim$  la relation d'équivalence :

$$s \sim t \quad \text{si et seulement si} \quad s \text{ et } t \text{ ont même type d'isomorphisme,}$$

pour toutes  $\mathbf{F}[U]$ -structures  $s$  et  $t$ .

Le type d'isomorphisme de  $\mathbf{F}$ -structures d'ordre  $n$  est la classe d'équivalence (modulo la relation  $\sim$ ) des  $\mathbf{F}$ -structures sur un ensemble fini  $U$  de cardinal  $n$ . On note  $T(\mathbf{F}_n)$  le quotient d'ensemble  $\mathbf{F}[n] / \sim$ . Les types d'isomorphismes de  $\mathbf{F}$ -structures sont généralement appelés les **objets non-étiquetés** d'une « classe combinatoire » dans [FS09, Chapitre I].

Dans de nombreux cas, on s'intéressera uniquement aux structures (objets étiquetés) ou uniquement aux types d'isomorphismes (objets non-étiquetés) d'une espèce  $\mathbf{F}$  décrits comme les ensembles  $\mathcal{C}_{\mathbf{F}}$  et  $\tilde{\mathcal{C}}_{\mathbf{F}}$  gradués, finis degré par degré, respectivement :

$$\mathcal{C}_{\mathbf{F}} = \sum_{n \geq 0} \mathbf{F}[n] \quad \text{ou} \quad \tilde{\mathcal{C}}_{\mathbf{F}} = \sum_{n \geq 0} T(\mathbf{F}[n]). \quad (1.15)$$

**Remarque 1.5.5 :** La notation de la classe combinatoire des structures non-étiquetés (ou à isomorphisme près),  $\tilde{\mathcal{C}}_{\mathbf{F}}$ , coïncide avec la notation des séries génératrices de ces structures non-étiquetés (cf. Définition 1.1.1).

Ces deux « types » d'ensembles gradués définissent respectivement la notion de **classe combinatoire étiquetée** (ou bien **étiquetée**) et de **classe combinatoire non-étiquetée** au sens du respectivement second et premier chapitres de [FS09].

**Exemple 1.5.6 :** Les *partitions d'ensembles* (§ 1.1.2) sont les structures (étiquetées) de l'espèce des partitions ; et, les *partitions d'entiers* (§ 1.1.1) sont les types d'isomorphismes associés. (L'action du groupe symétrique sur les partitions d'ensembles est l'action (définie sur les ensembles) :  $\sigma \cdot \{U, V, \dots\} = \{\sigma(U), \sigma(V), \dots\}$ .) On note **Par** l'espèce des partitions.

## 1.5.2. Séries associées

À chaque espèce  $\mathbf{F}$ , on peut associer trois différents types de séries :

### Série génératrice des structures

Une série génératrice exponentielle qui énumère les  $\mathbf{F}$ -structures suivant la cardinalité des ensembles finis :

$$\mathbf{F}(t) = \sum_{n \geq 0} \mathbf{f}_n \frac{t^n}{n!},$$

où  $\mathbf{f}_n$  est le nombre de  $\mathbf{F}$ -structures sur les ensembles de cardinalité  $n$ .

## Série génératrice des types d'isomorphismes

Une série génératrice ordinaire qui énumère les types d'isomorphismes de  $\mathbf{F}$ -structures (autrement dit les  $\mathbf{F}$ -structures à isomorphisme près) :

$$\tilde{F}(t) = \sum_{n \geq 0} \tilde{f}_n t^n,$$

où  $\tilde{f}_n$  est le nombre de classes d'équivalences modulo la relation  $\sim$  d'ordre  $n$ .

## Série indicatrice de cycles

Les *séries indicatrices de cycles* sont des séries plus générales que les deux précédentes. Ce sont des séries définies dans les fonctions symétriques. Par spécialisation, elles permettent de retrouver les deux précédentes.

**Définition 1.5.7 :** Soit  $U$  un ensemble fini et soit  $\sigma$  une permutation de  $U$ . Le *type cyclique* de  $\sigma$  est une séquence  $(\sigma^{(1)}, \sigma^{(2)}, \dots)$  où  $\sigma^{(k)}$  est le nombre de cycles de longueur  $k$  dans la décomposition en cycles disjoints de  $\sigma$ .

On définit les notations suivantes :

$$\text{Fix } \sigma = \{ u \in U \mid \sigma(u) = u \} \quad \text{et} \quad \text{fix } \sigma = \# \text{Fix } \sigma.$$

$\text{Fix } \sigma$  est l'ensemble des points fixes de  $\sigma$  ; on va plus particulièrement s'intéresser à  $\text{fix } \sigma = \sigma^{(1)}$  qui est le nombre de points fixes. On peut appliquer ces outils sur n'importe quelle espèce  $\mathbf{F}$ . À chaque permutation  $\sigma$  de  $U$ , on va associer  $\text{fix } \mathbf{F}[\sigma]$ , le nombre de points fixes associés à la permutation (de  $\mathbf{F}[U]$ ) définie par le transport de structures le long de  $\sigma$ .

**Remarque 1.5.8 :** La notation  $\text{Fix } \sigma$  est la même que  $\text{Fix } g$  (1.14) dans la section précédente.

**Définition 1.5.9 :** La *série indicatrice de cycles* d'une espèce  $\mathbf{F}$  est une série formelle (dans les fonctions symétriques) définie par :

$$Z_{\mathbf{F}}(p_1, p_2, p_3, \dots) = \sum_{n \geq 0} \left( \frac{1}{n!} \sum_{\sigma \in \mathcal{S}_n} \text{fix } \mathbf{F}[\sigma] p_1^{\sigma^{(1)}} p_2^{\sigma^{(2)}} p_3^{\sigma^{(3)}} \dots \right),$$

où  $(p_{\lambda})_{\lambda \vdash n}$  est la base des sommes de puissances de  $\text{Sym}$ .

Pour tout  $n \geq 0$ , la somme à l'intérieur (entre parenthèses) sur l'ensemble des permutations de degré  $n$  est le *caractère de Frobenius* des  $\mathbf{F}$ -structures sous l'action du groupe symétrique  $\mathcal{S}_n$ , noté  $\text{ch}(\mathbf{F}; n)$  (cf. § 1.4) :

$$\text{ch}(\mathbf{F}; n) = \frac{1}{n!} \sum_{\sigma \in \mathcal{S}_n} \text{fix } \mathbf{F}[\sigma] p_1^{\sigma^{(1)}} p_2^{\sigma^{(2)}} p_3^{\sigma^{(3)}} \dots$$

**Remarque 1.5.10** ([BLL98, Remarque 10]) :

$$\text{ch}(\mathbf{F}; n) = \sum_{\lambda \vdash n} \text{fix } \mathbf{F}[\sigma(\lambda)] \frac{p_\lambda}{z_\lambda},$$

où  $\sigma(\lambda)$  est une bijection (quelconque) de  $[n]$  ayant  $\lambda$  comme type cyclique (cf. Équation 1.11 pour la définition de  $z_\lambda$ ). Cette formule regroupe les permutations par type cyclique. (On rappelle que  $z_\lambda$  est définie (1.11).)

On peut retrouver les séries génératrices exponentielle et ordinaire associées aux structures et aux types d'isomorphismes par les spécialisations suivantes :

**Théorème 1.5.11** ([BLL98, Théorème 8]) :

$$\mathbf{F}(t) = Z_{\mathbf{F}}(t, 0, 0, \dots) \quad \text{et} \quad \widetilde{\mathbf{F}}(t) = Z_{\mathbf{F}}(t, t^2, t^3, \dots).$$

La première spécialisation est facile à déduire, elle correspond à extraire de la série indicatrice de cycle les termes  $\text{fix } \mathbf{F}[\sigma]$  où  $\sigma$  ne contient que des points fixes, autrement dit où  $\sigma$  est l'identité et donc fixe toutes les  $\mathbf{F}$ -structures. La seconde permet de retrouver exactement la formule du Lemme 1.4.6 de Burnside.

**Exemple 1.5.12** : La série indicatrice de cycle de l'espèce des permutations :

$$Z_{\mathcal{S}}(p_1, p_2, p_3, \dots) = \frac{1}{(1 - p_1)(1 - p_2)(1 - p_3) \dots},$$

D'où

$$\text{ch}(\mathcal{S}; n) = \sum_{\lambda \vdash n} p_\lambda.$$

PREUVE :

$$\begin{aligned} Z_{\mathcal{S}}(p_1, p_2, p_3, \dots) &= \prod_{i \geq 1} \frac{1}{1 - p_i} = \prod_{i \geq 1} \left( \sum_{j \geq 0} (p_i)^j \right) = \prod_{i \geq 1} \left( 1 + \sum_{j > 0} (p_i)^j \right), \\ &= 1 + \sum_{\substack{i_1 j_1 + i_2 j_2 + \dots + i_m j_m > 0 \\ 0 < i_1 < i_2 < \dots < i_m \\ j_k > 0, \text{ pour tout } 1 \leq k \leq m}} (p_{i_1})^{j_1} (p_{i_2})^{j_2} \dots (p_{i_m})^{j_m}, \\ &= \sum_{n \geq 0} \sum_{\lambda \vdash n} p_\lambda. \end{aligned} \quad \blacksquare$$

On déduit alors du théorème :

$$\mathcal{S}(t) = \frac{1}{1 - t} \quad \text{et} \quad \widetilde{\mathcal{S}}(t) = \frac{1}{(1 - t)(1 - t^2)(1 - t^3) \dots}.$$

La première spécialisation du théorème provient de la spécialisation suivante des fonctions symétriques  $p_1 \mapsto t$  et  $p_n \mapsto 0$  si  $n > 1$ . La seconde correspond à  $p_n \mapsto t^n$ . Ces dernières correspondent dans la base  $(h_\lambda)$  des fonctions complètes de  $\text{Sym}$  à :

$$h_\lambda \mapsto \binom{n}{\lambda_1, \dots, \lambda_k} t^n \quad \text{et} \quad h_\lambda \mapsto t^n,$$

pour toute partition  $\lambda = \lambda_1 \dots \lambda_k$  de l'entier  $n$ .

### 1.5.3. Espèces caractéristiques

Voici quelques exemples d'espèces caractéristiques qu'on utilisera par la suite pour en définir de nombreuses autres à partir des opérateurs sur les espèces. Soient  $U$  et  $V$  deux ensembles finis et soit  $\sigma : U \rightarrow V$  une bijection.

#### Ensembles $E$

L'espèce des ensembles  $E$  se définit simplement par :

$$E[U] = \{U\} \quad \text{et} \quad E[\sigma] = U \mapsto \sigma(U).$$

Comme  $E[U]$  est un singleton, il est alors fixé par toute les bijections. On en déduit alors facilement la série indicatrice de cycle :

$$Z_E(p_1, p_2, p_3, \dots) = \sum_{n \geq 0} \sum_{\lambda \vdash n} \frac{p_\lambda}{z_\lambda} = \sum_{n \geq 0} h_n = \sigma_1(\mathbb{X}).$$

(La formule dans la base des fonctions homogènes se déduit immédiatement du changement de base (1.10) et  $\sigma_t(\mathbb{X})$  est définie (1.9).) Par conséquent les séries des structures et types d'isomorphismes :

$$E(t) = \exp(t) \quad \text{et} \quad \tilde{E}(t) = \frac{1}{1-t}.$$

#### Zéro $0$ , Un $1$ et Singletons $X$

Les espèces  $0$ ,  $1$  et  $X$  sont les trois espèces qu'on associe naturellement aux séries  $0$ ,  $1$  et  $t$  :

– Zéro :

$$0[U] = \emptyset \quad \text{avec} \quad Z_0(p_1, p_2, p_3, \dots) = 0.$$

– Un :

$$1[U] = \begin{cases} \{U\} & \text{si } U = \emptyset, \\ \emptyset & \text{sinon;} \end{cases} \quad \text{avec} \quad Z_1(p_1, p_2, p_3, \dots) = 1.$$

– Singletons :

$$X[U] = \begin{cases} \{U\} & \text{si } \#U = 1, \\ \emptyset & \text{sinon;} \end{cases} \quad \text{avec} \quad Z_X(p_1, p_2, p_3, \dots) = p_1.$$

### 1.5.4. Somme, Produit, Composition et Équation fonctionnelle

La puissance de la théorie des espèces réside naturellement dans la pléthore d'opérateurs sur les structures : *somme*, *produit de Cauchy*, *produit cartésien*, *substitution plethystique*, *composition fonctorielle*, *pointage*, *dérivation* (ou encore *produit arithmétique* [MM08]). Chacun de ces opérateurs peut-être considérés comme une traduction combinatoire d'une opération « naturelle » (du même nom) sur les séries formelles. Ces opérations permettent aussi de définir récursivement les espèces, comme l'espèce des arbres binaires ou encore l'espèce des graphes simples.

#### Somme

Soient  $F$  et  $G$  deux espèces de structures.

**Définition 1.5.13 :** La *somme* des espèces  $F$  et  $G$  est l'espèce  $F + G$  définie par : l'ensemble des  $(F + G)$ -structures sur  $U$  est l'union disjointe des  $F$ -structures et des  $G$ -structures sur cet ensemble,

$$(F + G)[U] := F[U] \sqcup G[U] .$$

Le transport le long d'une bijection  $\sigma : U \rightarrow V$  est

$$(F + G)[\sigma] : \begin{cases} s \in F[U] & \mapsto F[\sigma](s) , \\ s \in G[U] & \mapsto G[\sigma](s) . \end{cases}$$

Les séries associées aux structures, aux types d'isomorphismes et la série indicatrice de  $F + G$  sont respectivement :

$$(F + G)(t) = F(t) + G(t) , \quad \widetilde{(F + G)}(t) = \widetilde{F}(t) + \widetilde{G}(t) , \quad Z_{F+G} = Z_F + Z_G .$$

La somme de deux espèces s'étend naturellement à une famille d'espèce  $(F_i)_{i \in I}$ . Une famille d'espèces  $(F_i)_{i \in I}$  est *sommable* si pour tout ensemble fini  $U$ , il existe un nombre fini d'indices  $i \in I$  tel que  $F_i[U] \neq \emptyset$ . (Les structures et séries associées sont définies comme attendu par l'union disjointe des structures et la somme des séries.)

**Définition 1.5.14 :** Toute espèce  $F$  se décompose de manière canonique en la somme de la famille d'espèces  $(F_n)_{n \geq 0}$  définies par :

$$F_n[U] = \begin{cases} F[U] & \text{si } \#U = n, \\ \emptyset & \text{sinon ;} \end{cases} \quad \text{pour tout ensemble fini } U ,$$

et où le transport de structures reste identique à celui de  $F$ . L'espèce  $F_n$  est la *restriction de  $F$  aux ensembles de cardinalité  $n$* .

On note aussi  $F_+$  la restriction de  $F$  aux ensembles non-vides :

$$F_+[U] = \begin{cases} \emptyset & \text{si } U = \emptyset, \\ F[U] & \text{sinon.} \end{cases}$$

**Remarque 1.5.15 :** L'espèce  $0$  est l'élément neutre pour la somme d'espèces.

## Produit de Cauchy

**Définition 1.5.16 :** Le **produit (de Cauchy)** des espèces  $F$  et  $G$  est l'espèce  $F \cdot G$  définie par : l'ensemble des  $(F \cdot G)$ -structures sur  $U$  est l'union disjointe des produits cartésiens des  $F$ -structures et des  $G$ -structures sur toutes les décompositions de  $U$  en  $S$  et  $T$ ,

$$(F \cdot G)[U] = \sum_{S \sqcup T = U} F[S] \times G[T].$$

Le transport le long d'une bijection  $\sigma : U \rightarrow V$  est

$$(F \cdot G)[\sigma](s) = (F[\sigma|_S](t), G[\sigma|_T](u)),$$

pour toute  $(F \cdot G)$ -structure  $s = (t, u)$  sur  $U$ , avec  $\sigma|_S$  (resp.  $\sigma|_T$ ) la restriction de  $\sigma$  à  $S$  (resp.  $T$ ).

Les séries associées sont respectivement le produit des séries :

$$(F \cdot G)(t) = F(t) G(t), \quad (\widetilde{F \cdot G})(t) = \widetilde{F}(t) \cdot \widetilde{G}(t), \quad Z_{F \cdot G} = Z_F Z_G.$$

où respectivement les coefficients sont :

$$\begin{aligned} [t^n](F \cdot G)(t) &= \sum_{i+j=n} \binom{n}{i} [t^i]F(t) [t^j]G(t), \\ [t^n](F \cdot G)(t) &= \sum_{i+j=n} [t^i]F(t) [t^j]G(t), \\ \text{ch}(F \cdot G; n) &= \sum_{i+j=n} \text{ch}(F; i) \text{ch}(G; j). \end{aligned}$$

Tout comme pour la somme, le produit des espèces se généralise aux familles finies d'espèces  $(F_i)_{1 \leq i \leq k}$  : les  $(F_i)_{1 \leq i \leq k}$ -structures,

$$(F_1 \cdot F_2 \cdots F_k)[U] = \sum_{U_1 \sqcup U_2 \sqcup \cdots \sqcup U_k = U} F_1[U_1] \times F_2[U_2] \times \cdots \times F_k[U_k],$$

et le transport de structures le long de la bijection  $\sigma : U \rightarrow V$ ,

$$(F_1 \cdot F_2 \cdots F_k)[\sigma](s) = (F_1[\sigma|_{U_1}](s_1), F_2[\sigma|_{U_2}](s_2), \cdots, F_k[\sigma|_{U_k}](s_k)),$$

pour toute  $(F_i)_{1 \leq i \leq k}$ -structure  $s = (s_1, s_2, \cdots, s_k)$ , avec  $\sigma|_{U_i}$  est la restriction de  $\sigma$  à  $U_i$  pour tout  $1 \leq i \leq k$ .

**Remarque 1.5.17 :** L'espèce  $1$  est l'élément neutre pour le produit d'espèces.

## Composition partitionnelle

Soit **Par** l'espèce des partitions (Exemple 1.5.6).

**Définition 1.5.18 :** À la condition que  $G[\emptyset] = \emptyset$ , la **composition (partitionnelle)** de **G** dans **F**, noté  $F(G)$  ou  $F \circ G$ , est définie par : l'ensemble des  $(F \circ G)$ -structures sur  $U$  est l'ensemble des triplets  $s = (\pi, t, u)$  où

1.  $\pi$  est une partition de  $U$ ,
2.  $t$  est une **F**-structure sur l'ensemble des parts de  $\pi$ ,
3.  $u = (u_p)_{p \in \pi}$  est une famille de **G**-structures où chaque  $u_p$  est une **G**-structure sur  $p$ .

C'est-à-dire pour tout  $U$ ,

$$(F \circ G)[U] := \sum_{\pi \vdash U} F[\pi] \times \prod_{p \in \pi} G[p],$$

$$\left( \simeq \sum_{\pi \vdash U} \{\pi\} \times F[\pi] \times \prod_{p \in \pi} G[p], \right)$$

et pour toute bijection  $\sigma : U \rightarrow V$ , le transport de structure est donnée par :

$$(F \circ G)[\sigma](s) := \left( \text{Par}[\sigma](\pi), F[\text{Par}[\sigma]](t), (G[\sigma|_p](u_p))_{p \in \pi} \right).$$

Les séries associées sont respectivement :

$$(F \circ G)(t) = F(G(t)), \quad (\widetilde{F \circ G})(t) = Z_F(\widetilde{G}(t), \widetilde{G}(t^2), \widetilde{G}(t^3), \dots),$$

et

$$Z_{F \circ G}(p_1, p_2, p_3, \dots) = Z_F(Z_G(p_1, p_2, p_3, \dots), Z_G(p_2, p_4, p_6, \dots), Z_G(p_3, p_6, p_9, \dots)).$$

Cette dernière opération sur les séries indicatrices de cycles correspond au **pléthysme** dans les fonctions symétriques :

$$Z_{F \circ G}(p_1, p_2, p_3, \dots) = \text{ch}(F, 0) + \sum_{n \geq 1} \sum_{\lambda \vdash n} \text{fix } F[\sigma(\lambda)] \frac{p_\lambda [Z_G(p_1, p_2, \dots)]}{z_\lambda}.$$

## Solution d'équations fonctionnelles

À partir de ces opérateurs (et d'autres non décrits ici), une manière concise de décrire une espèce consiste à la définir par une équation fonctionnelle. L'exemple le plus connu est certainement celui de l'espèce des arbres binaires décrite par l'équation :

$$\mathcal{B} = 1 + \mathcal{B} \cdot X \cdot \mathcal{B}, \quad (1.16)$$

qui décrit une structure d'arbre binaire comme soit une feuille **1**, soit un triplet  $(g, x, d)$  où  $g$  et  $d$  sont respectivement les sous-arbres gauche et droite d'un arbre binaire de

racine  $x$ . De manière explicite, l'ensemble des structures sur  $U$  et le transport de structures le long d'une bijection  $\sigma$  se déduisent à partir des définitions des différentes espèces (dîtes terminales)  $\mathbf{1}$  et  $\mathbf{X}$  et de celles des opérateurs  $+$  et  $\cdot$  :

$$\begin{aligned}\mathcal{B}[U] &= \mathbf{1}[U] \sqcup (\mathcal{B} \cdot \mathbf{X} \cdot \mathcal{B})[U], \\ &= \mathbf{1}[\emptyset], \quad \text{si } U = \emptyset \quad \text{et} \quad \sum_{\substack{R \sqcup S \sqcup T = U \\ \#S=1}} \mathcal{B}[R] \times \mathbf{X}[S] \times \mathcal{B}[T] \quad \text{sinon.}\end{aligned}$$

$$\mathcal{B}[\sigma](t) = t \quad \text{si } t \in \mathcal{B}[\emptyset] \quad \text{et} \quad (\mathcal{B}[\sigma|_R](l), S, \mathcal{B}[\sigma|_T](r)) \quad \text{sinon.}$$

Ces définitions en termes d'équations fonctionnelles permettent de décrire de manière concise les objets combinatoires. Voici quelques exemples classiques que nous réutiliserons par la suite :

- l'espèce des *ordres linéaires* :

$$\mathcal{L} = \mathbf{1} + \mathbf{X} \cdot \mathcal{L}, \quad (1.17)$$

autrement dit, un ordre linéaire est soit une séquence vide (l'unique structure de l'espèce  $\mathbf{1}$ ) soit un symbole (c'est-à-dire une structure de singleton,  $\mathbf{X}$ ) concaténé(e) avec un ordre linéaire linéaire « plus petit » (récursivement une structure de  $\mathcal{L}$ ). À partir de cette équation, on déduit la série indicatrice de cycle :

$$Z_{\mathcal{L}} = \frac{1}{1 - h_1} = \sum_{n \geq 0} h_1^n. \quad (1.18)$$

- l'espèce des *arbres enracinés* :

$$\mathcal{A} = \mathbf{X} \cdot \mathbf{E}(\mathcal{A}), \quad (1.19)$$

ce qui correspond à dire qu'un arbre enraciné est une racine ( $\mathbf{X}$ ) à laquelle est greffée un ensemble (potentiellement vide) d'arbres enracinés ( $\mathbf{E}(\mathcal{A})$ ).

- l'espèce des *forêts d'arbres enracinés* :

$$\mathcal{F} = \mathbf{E}(\mathcal{A}) = \mathbf{E}(\mathbf{X} \cdot \mathcal{F}). \quad (1.20)$$

Une forêt d'arbres enracinés est un ensemble (non-ordonné) d'arbres enracinés. En remplaçant l'espèce des ensembles par l'espèces des ordres linéaires dans (1.20), on définit l'espèce des forêts ordonnées d'arbres (non-ordonnés) enracinés.

- l'espèce des *arbres enracinés ordonnés* :

$$\mathcal{A}_{\mathcal{L}} = \mathbf{X} \cdot \mathcal{L}(\mathcal{A}_{\mathcal{L}}).$$

- l'espèce des *partitions* (cf. Remarque 1.5.19) :

$$\mathbf{Par} = \mathbf{E}(\mathbf{E}_+). \quad (1.21)$$

- l'espèce des *compositions* :

$$\mathbf{Bal} = \mathcal{L}(\mathbf{E}_+) \quad (1.22)$$

**Remarque 1.5.19 :** L'espèce des partitions (Exemple 1.5.6) est isomorphe à l'espèce définie par l'équation (1.21). L'opérateur de composition (Définition 1.5.18) n'est pas correctement défini sinon.



**Première partie .**

**Autour des fonctions de parking**

Numbers written on restaurant bills within the confines of restaurants do not follow the same mathematical laws as numbers written on any other pieces of paper in any other parts of the Universe. This single fact took the scientific world by storm.

(Douglas ADAMS, *Life, the Universe and Everything*)

Les *fonctions de parking* ont été introduites par Alan G. KONHEIM et Benjamin WEISS dans [KW66, §6] pour modéliser un *problème de hachage* :

A car occupied by a man and his dozing wife enters the street on the left and moves towards the right. The wife awakens at a capricious moment and orders her husband to park immediately! He dutifully parks at his present location, if it is empty, and if not, continues to the right and parks at the next available space.

Traduit en termes informatiques, les *femmes* jouent le rôle d'une *fonction de hachage*. Dans chaque voiture, la femme qui l'occupe désigne un emplacement privilégié où garer la voiture. Autrement dit, une voiture est une clef à stocker, un emplacement de parking est une position dans la table de hachage et l'« ordre » donné par chaque femme définit l'image de la position préférentielle de la clef dans la table hachage. Le rôle des hommes correspond à l'implémentation de la *table de hachage* où sont gérés les conflits entre les clefs. Si la position préférentielle est libre alors la clef est stockée à cette position, sinon elle est stockée à la prochaine position libre.

Ce problème, foncièrement emprunt de la misogynie caractéristique de nombreux ouvrages des années soixante, a introduit une structure combinatoire riche qui apparaît dans de nombreux autres domaines : les *fonctions de parking*.

**Définition 1.5.20 :** Soit  $U$  un ensemble fini.

Une *fonction de parking* (sur  $U$ ) est une fonction  $f : U \rightarrow \mathbb{N}_{>0}$  telle que

$$\#\{f(u) \leq k \mid u \in U\} \geq k, \quad \text{pour tout } 1 \leq k \leq \#U. \quad (1.23)$$

L'ensemble  $U$  correspond aux voitures qui souhaitent se garer dans la rue, autrement dit à l'ensemble des clefs ; la fonction  $f$  correspond à l'affectation des places de parking pour chaque voiture, c'est-à-dire à la fonction de hachage ; et enfin, la *contrainte de parking* (1.23) permet de vérifier que chaque voiture aura une place, en termes de hachage cela signifie que cette implémentation de table de hachage *par adresse* est consistante sur l'ensemble de clefs  $U$ .

Ces structures sont en bijection avec les forêts d'arbres enracinés (1.19) et il existe de nombreuses bijections dont quelques-unes seront présentées Chapitre 5. Elles sont aussi en bijection avec les séquences de Prüfer et apparaissent dans différents contextes : arrangements d'hyperplans, partitions planes, polynômes de Gončarov, etc.. Elles occupent une part importante dans les recherches autour du théorème  $n!$  conjecturé par Adriano M. GARCIA et Mark D. HAIMAN [GH96] et prouvé par la suite par ce dernier [Hai01], à propos des dimensions des modules bigradués des polynômes harmoniques diagonaux.

Dans le premier chapitre de cette partie ([Chapitre 2](#)), on commence par rappeler l'espèce des séquences de  $k$  ensembles :  $\mathbf{E}^k$ . Celle-ci est très bien connue, très simple et surtout riche combinatoirement. Cette espèce et les formules d'énumération associées sont les outils principaux des deux sections suivantes.

On l'utilisera pour (re)définir les fonctions de parking dans le second chapitre ([Chapitre 3](#)) où on rappelle quelques résultats classiques sur les *fonctions de parking* [[KW66](#) ; [FR74](#) ; [Rio69](#)] et surtout [[NT08](#)] où les auteurs, Jean-Christophe NOVELLI et Jean-Yves THIBON établissent le caractère de Frobenius non-commutatif des fonctions de parking sous l'action de l'algèbre 0-Hecke.

Dans le troisième chapitre ([Chapitre 4](#)), on introduit les *fonctions de parking généralisées*. Dans un premier temps, nous rappellerons brièvement les travaux de Richard P. STANLEY et Jim PITMAN ayant introduit ces objets en calculant le volume de polyèdres de dimension  $n$  [[PS02](#)]. Nous rappellerons aussi quelques résultats de Joseph P.S. KUNG et Catherine H. YAN où ils énumèrent ces structures à partir de polynômes de Gončarov [[KY03](#)]. Dans la suite de ce chapitre on présentera les travaux réalisés avec A. VIRMAUX où les *fonctions de parking généralisées* sont redéfinies récursivement, en tant qu'espèce, par une équation fonctionnelle [[PV15](#)]. À partir de la théorie des espèces, on associe automatiquement une série indicatrice de cycle à cette définition constructive et récursive. On verra que le caractère de Frobenius (commutatif) pour l'action du groupe symétrique associée se remonte naturellement dans les fonctions symétriques non-commutatives comme le *caractère de Frobenius non-commutatif* pour l'action de l'algèbre 0-Hecke. Ce nouveau caractère révèle de nombreuses nouvelles formules d'énumérations dont l'expressivité nous sera très utile pour l'énumération des automates acycliques ([Chapitre 8](#)).

La grammaire développée dans le troisième chapitre met en avant une construction plus générale introduite dans le chapitre suivant ([Chapitre 5](#)). Nous profiterons de ce chapitre pour rappeler quelques bijections entre les fonctions de parking (généralisées) et les *forêts d'arbres enracinés* et certaines extensions de structures arborescentes (§ 5.2). Ces bijections vont nous permettre de décrire facilement les séries génératrices exponentielles de certaines de nos nouvelles structures de parking par des équations fonctionnelles.

## 2. Espèce des séquences finies d'ensembles

Les fonctions de parking sont des fonctions d'un ensemble fini  $U$  sur les entiers. Chaque fonction entière  $f : U \rightarrow \mathbb{N}_{>0}$  (plus généralement sur  $\mathbb{N}$ ) peut se définir de manière équivalente comme la séquence (finie) de ses images réciproques, autrement dit comme une séquence d'ensembles disjoints. Dans cette section, on rappelle l'espèce des séquences de  $k$  ensembles,  $\mathbf{E}^k$ , pour redéfinir les fonctions de parking [PV15]. Cette espèce simple nous permet de retrouver facilement la plupart des résultats connus sur les fonctions de parking, en particulier les formules d'énumération dues à Henri O. POLLAK. Elle nous est aussi très utile dans la dernière section de ce chapitre pour décrire les fonctions de parking généralisées à partir d'une grammaire.

En utilisant le produit de Cauchy des espèces on définit  $\mathbf{E}^k$ , l'espèce des séquences de  $k$  ensembles (disjoints) :

$$\mathbf{E}^k = \mathbf{E} \cdot \mathbf{E}^{k-1} \quad \text{avec} \quad \mathbf{E}^1 = \mathbf{E} \quad \text{et} \quad \mathbf{E}^0 = 1.$$

**Remarque 2.0.21 :** Dans le cas où  $U = [n]$ , on peut alors associer de manière canonique un mot sur l'alphabet  $[k]$  à toute séquence  $(Q_i)$  de  $\mathbf{E}^k[n]$  par la bijection suivante :

$$w_1 \cdots w_n \longleftrightarrow (Q_i)_{1 \leq i \leq k} \quad \text{avec} \quad Q_i := \{j \in [n] \mid w_j = i\}. \quad (2.1)$$

**Exemple 2.0.22 :** La séquence  $(\{1, 3\}, \{\}, \{\}, \{2\}, \{4\})$  correspond au mot 1415 (de longueur 4 sur l'alphabet  $[5]$ ).

**Exemple 2.0.23 :** Les  $\mathbf{E}^3$ -structures sur  $\{a, b\}$  correspondent aux neuf séquences :

$$(ab \mid \cdot \mid \cdot), (\cdot \mid ab \mid \cdot), (\cdot \mid \cdot \mid ab), (a \mid b \mid \cdot), \\ (a \mid \cdot \mid b), (\cdot \mid a \mid b), (b \mid a \mid \cdot), (b \mid \cdot \mid a), (\cdot \mid b \mid a),$$

où  $(\alpha\beta\cdots \mid \gamma\cdots \mid \dots)$  dénote la séquence  $(\{\alpha, \beta, \dots\}, \{\gamma, \dots\}, \dots)$  et où  $\cdot$  dénote l'ensemble vide.

### 2.1. Séquences et chemins décorés

Dans la suite, les propriétés sur les fonctions de parking généralisées pourront se traduire sur des chemins décorés. Nous utiliserons ces chemins pour véhiculer l'intuition d'une manière simple et graphique. Cette notion de chemin décoré peut déjà se définir sur n'importe quelle séquence d'ensembles.

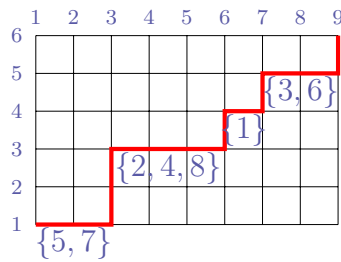
Soit  $(Q_i)$  une séquence de  $n$  ensembles.

**Définition 2.1.1 :** Le **chemin décoré** de  $(Q_i)$  est un chemin (en escalier) dans le plan composé uniquement de pas horizontaux et verticaux, respectivement appelés **marche** et **contre-marche**. Pour tout  $1 < i \leq n$ , l'ensemble  $Q_i$  décore la marche de longueur  $\#Q_i$  à hauteur  $i$  consécutive à la marche à hauteur  $i - 1$  décorée par  $Q_{i-1}$ , la première marche à hauteur 1 est décorée par  $Q_1$  et débute à la coordonnée d'abscisse 1.

**Remarque 2.1.2 :** Si  $Q_i = \emptyset$  alors la marche associée est de longueur 0.

Pour ne pas surcharger les chemins, on choisit de ne pas représenter les décorations des marches associées aux ensembles vides.

**Exemple 2.1.3 :** La séquence  $(57 \mid \cdot \mid 248 \mid 1 \mid 36 \mid \cdot)$  est représentée par le chemin décoré :



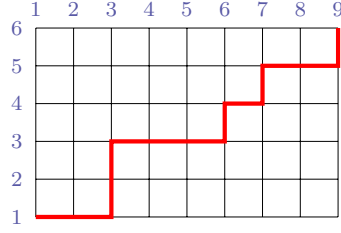
**Remarque 2.1.4 :** Le choix de débiter le chemin aux coordonnées  $(1, 1)$  est une convention qui prendra sens dans la section sur les fonctions de parking généralisées (Chapitre 4).

**Remarque 2.1.5 :** Le chemin décoré d'une séquence de  $k$  ensembles sur un ensemble à  $n$  éléments est un chemin d'origine  $(1, 1)$  jusqu'à  $(n + 1, k)$ .

## 2.2. Types d'isomorphismes

Le **type d'isomorphisme** d'une séquence  $(Q_i)$  de  $k$  ensembles (sur  $U$  un ensemble à  $n$  éléments) peut s'identifier à une séquence d'entiers (positif ou nul)  $(q_i)$  (de somme  $n$ ) définie par  $q_i := \#Q_i$ . On parle alors d'une **décomposition** de  $n$  en  $k$  parts (cf. Remarque 1.1.9). En effet, il est facile de vérifier que toute séquence d'ensembles  $(Q'_i)$  sur  $U$  a même type d'isomorphisme que  $(Q_i)$  si et seulement si elle s'envoie sur la décomposition  $(q_i)$ . Le type d'isomorphisme d'un chemin décoré correspond donc au chemin sans décoration.

**Exemple 2.2.1 :** Le type d'isomorphisme de la séquence de l'exemple précédent (Exemple 2.1.3) est la décomposition de 8 en 6 parts  $(2, 0, 3, 1, 2, 0)$ . Ceci correspond au chemin-escalier dont la première marche est de longueur 2, la suivante de longueur 0, puis 3, 1, etc. :



Le type d'isomorphisme de la séquence  $(13 \mid \cdot \mid 2 \mid \cdot \mid \cdot \mid 4)$  de 6 ensembles sur [4] est la décomposition  $(2, 0, 1, 0, 0, 1)$  de l'entier 4 en 6 parts.

## 2.3. Séries génératrices

Le caractère  $h_n(k \cdot \mathbb{X})$  est bien connu pour être le caractère de Frobenius des mots de longueur  $k$  sur un alphabet de  $n$  symboles sous l'action du groupe symétrique. En termes d'espèces, ce caractère correspond à celui des séquences de  $k$  ensembles sur un ensemble de cardinalité  $n$  :

$$\text{ch}(\mathbb{E}^k; n) = h_n(k \cdot \mathbb{X}) \quad (2.2)$$

D'après la définition, on retrouve ce résultat à partir de la série indicatrice de l'espèce des ensembles (§ 1.5.3) :

$$Z_{\mathbb{E}^k} = (Z_{\mathbb{E}})^k = \left( \sum_{n \geq 0} h_n \right)^k,$$

en développant le produit des séries, on retrouve une somme sur toutes les partitions :

$$Z_{\mathbb{E}^k} = \sum_{n \geq 0} \sum_{\lambda \vdash n} \binom{\ell(\lambda)}{\lambda^{(1)}, \dots, \lambda^{(n)}} \binom{k}{\ell(\lambda)} h_{\lambda},$$

où le binomial  $\binom{k}{\ell(\lambda)}$  correspond à choisir les  $\ell(\lambda)$  ensembles non-vides (où se répartissent les étiquettes) parmi les  $k$  possibles. Les parts de  $\lambda$  désignent les cardinaux de ces ensembles non-vides. Le multinomial  $\binom{\ell(\lambda)}{\lambda^{(1)}, \dots, \lambda^{(n)}}$  correspond à fixer l'ordre des ensembles non-vides en fonctions des parts de  $\lambda$  ;

$$Z_{\mathbb{E}^k} = \sum_{n \geq 0} \sum_{\lambda \vdash n} m_{\lambda}(k) h_{\lambda}, \quad [\text{Éq. 1.12}]$$

$$= \sum_{n \geq 0} h_n(k \cdot \mathbb{X}) = \sigma_1(k \cdot \mathbb{X}). \quad [\text{Éq. 1.9}]$$

On déduit immédiatement, par spécialisation (Théorème 1.5.11), les séries génératrices exponentielle et ordinaire :

$$\mathbb{E}^k(t) = (\mathbb{E}(t))^k = e^{kt} = \sum_{n \geq 0} k^n \frac{t^n}{n!}, \quad (2.3)$$

$$\tilde{\mathbf{E}}^k(t) = \left(\tilde{\mathbf{E}}(t)\right)^k = \frac{1}{(1-t)^k} = \sum_{n \geq 0} \binom{n+k-1}{k-1} t^n, \quad (2.4)$$

et les identités suivantes :

$$\sum_{\substack{\lambda \vdash n \\ \lambda = \lambda_1 \cdots \lambda_m}} \binom{\ell(\lambda)}{\lambda^{(1)}, \dots, \lambda^{(n)}} \binom{k}{\ell(\lambda)} \binom{n}{\lambda_1, \dots, \lambda_m} = k^n,$$

$$\sum_{\lambda \vdash n} \binom{\ell(\lambda)}{\lambda^{(1)}, \dots, \lambda^{(n)}} \binom{k}{\ell(\lambda)} = \binom{n+k-1}{k-1}.$$

## 2.4. Relèvement non-commutatif

Ce caractère (2.2) se relève naturellement, dans l'algèbre de Hopf des fonctions symétriques non-commutatives :

$$\begin{aligned} \mathbf{ch}(\mathbf{E}^k; n) &= \mathbf{S}_n(k \cdot \mathbb{A}) = \sum_{\pi \models n} \mathbf{M}_\pi(k) \mathbf{S}^\pi = \sum_{\pi \models n} \binom{k}{\ell(\pi)} \mathbf{S}^\pi, \\ &= \mathbf{R}_n(k \cdot \mathbb{A}) = \sum_{\pi \models n} \mathbf{F}_\pi(k) \mathbf{R}_\pi = \sum_{\pi \models n} \binom{k-1+\ell(\pi^\sim)}{n} \mathbf{R}_\pi, \\ &= \sum_{\pi \models n} (-1)^{n-\ell(\pi)} \Lambda^\pi(k \cdot \mathbb{A}) = \sum_{\pi \models n} (-1)^{n-\ell(\pi)} \binom{k-1+\ell(\pi)}{\ell(\pi)} \Lambda^\pi. \end{aligned} \quad (2.5)$$

Il est intéressant de noter que cette série développée dans différentes bases des fonctions symétriques non-commutatives correspond encore à un caractère en théorie des représentations. Il s'agit du caractère de Frobenius (non-commutatif) de la représentation des séquences de  $k$  ensembles sous l'action de l'algèbre de Hecke dégénérée (à  $q = 0$ ) [KLT97].

Ce relèvement est très intéressant combinatoirement. Il va nous permettre d'extraire des informations beaucoup plus fines que dans le cas des fonctions symétriques. Dans la section suivante, nous traiterons de l'interprétation des coefficients dans chacune des bases des fonctions symétriques non-commutatives.

## 2.5. Interprétation de $\mathbf{S}^\pi$

Dans la base des fonctions complètes, le coefficient  $\alpha_\pi$  du terme  $\mathbf{S}^\pi$  (dans le caractère  $\mathbf{ch}(\mathbf{E}^k; n)$  avec  $\pi = \pi_1 \pi_2 \cdots \pi_k$ ) correspond au nombre de séquences d'ensembles  $(Q_i)$  (à isomorphisme près) telles que le premier ensemble non-vidé de  $(Q_i)$  contient  $\pi_1$  éléments, le second non-vidé contient  $\pi_2$  et ainsi de suite jusqu'au  $k^{\text{ième}}$  et dernier ensemble non-vidé contenant  $\pi_k$  éléments. (La base  $(\mathbf{S}^\pi)$  des fonctions complètes est un relèvement non-commutatif de la base des fonctions homogènes  $(h_\lambda)$ . Ce relèvement permet ainsi d'interpréter le produit cartésien d'ensembles directement sur le caractère des séquences d'ensembles.)

En termes de chemin décoré (Définition 2.1.1), la composition  $\pi$  définit à la fois la longueur de toutes les marches de longueur non-nulle et leurs ordres d'apparition : la première marche de longueur non-nulle est de longueur  $\pi_1$ , la seconde de longueur  $\pi_2$ , etc. jusqu'à la  $k^{\text{ième}}$  et dernière de longueur  $\pi_k$ .

**Exemple 2.5.1 :** Le caractère de Frobenius non-commutatif de  $\mathbb{E}^4$  pour les ensembles de cardinalité 5 sous l'action de  $H_n(0)$  :

$$\text{ch}(\mathbb{E}^5; 4) = 5\mathbf{S}_{1111} + 10\mathbf{S}_{112} + 10\mathbf{S}_{121} + 10\mathbf{S}_{13} + 10\mathbf{S}_{211} + 10\mathbf{S}_{22} + 10\mathbf{S}_{31} + 5\mathbf{S}_4, \quad (2.6)$$

où on a bien 10 séquences de 4 ensembles (à isomorphisme près) telles que les premier et le second ensembles non-vides soient des singletons et le suivant et dernier non-vide contient 2 éléments :

$$(1, 1, 2, 0, 0), (1, 1, 0, 2, 0), (1, 0, 1, 2, 0), (0, 1, 1, 2, 0), (1, 1, 0, 0, 2), \\ (1, 0, 1, 0, 2), (0, 1, 1, 0, 2), (1, 0, 0, 1, 2), (0, 1, 0, 1, 2), (0, 0, 1, 1, 2).$$

Cette interprétation, dans les fonctions complètes, définit un partitionnement de l'ensemble des séquences d'ensembles en fonctions des compositions. À chaque séquence, on associe une composition à partir de son évaluation :

**Définition 2.5.2 :** Soit  $U$  un ensemble de  $n$  éléments. L'évaluation tassée d'une séquence (finie) d'ensembles  $s = (Q_i)$  sur  $U$  est la composition  $\pi \models n$  définie par :

$$\pi := (\#Q_i \mid i \in [n] \text{ et } Q_i \neq \emptyset),$$

avec  $i$  qui varie de 1 à  $\ell(s)$  dans l'ordre naturel.

On note  $\text{pEv}(s)$  l'évaluation tassée de  $s$ . On alors que le nombre de séquences de  $k$  ensembles ayant pour évaluation tassée  $\pi$  correspond au nombre obtenu par la spécialisation de la quasi-monomiale  $\mathbf{M}_\pi$  :

$$\mathbf{M}_\pi(k) = \#\{s \in \mathbb{E}^k[n] \mid \text{pEv}(s) = \pi\},$$

pour tout  $k \in \mathbb{N}_{>0}$  et toute composition  $\pi \models n$ .

## 2.6. Interprétation de $\mathbf{R}_\pi$

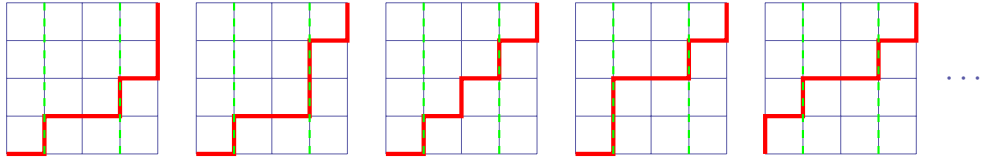
Dans la base des fonctions rubans, le coefficient  $\alpha_\pi$  du terme  $\mathbf{R}_\pi$  (dans le caractère  $\text{ch}(\mathbb{E}^k; n)$  avec  $\pi = \pi_1\pi_2 \cdots \pi_m$ ) s'interprète facilement en termes de chemin (décoré à isomorphisme près) comme les chemins de longueur  $n$  et de hauteur  $k$  avec des contremarches forcées aux coordonnées d'ordonnée  $\pi(j)$  pour tout  $1 \leq j < m$ .

**Exemple 2.6.1 :** En reprenant l'Exemple 2.5.1, l'expression du caractère des séquences de 5 ensembles sur les ensembles de cardinalité 4 dans la base des fonctions rubans est :

$$\text{ch}(\mathbb{E}^5; 4) = 5\mathbf{R}_{1111} + 15\mathbf{R}_{112} + 15\mathbf{R}_{121} + 35\mathbf{R}_{13} + 15\mathbf{R}_{211} + 35\mathbf{R}_{22} + 35\mathbf{R}_{31} + 70\mathbf{R}_4 \quad (2.6')$$

Voici quelques-uns des 15 chemins de longueur 4 et de hauteur 5 associés au terme  $\mathbf{R}_{121}$  (où les coordonnées des contremarches forcées sont représentées par une droite hachée verte) :





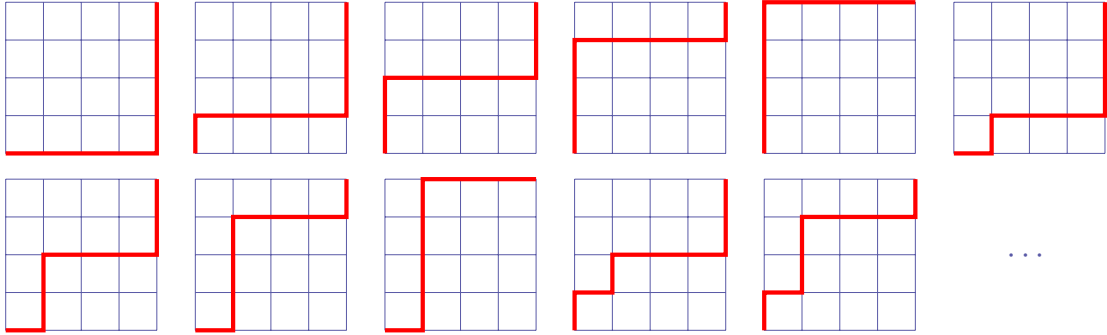
## 2.7. Interprétation de $\Lambda^\pi$

Dans la base des fonctions élémentaires, le coefficient  $\alpha_\pi$  (au signe près) du terme  $\Lambda^\pi$  (du caractère  $\mathbf{ch}(\mathbf{E}^k; n)$ ) correspond au nombre de chemin de longueur  $n$  et de hauteur  $k$  dont l'évaluation tassée (Définition 2.5.2)  $\tau$  est plus grossière que  $\pi$  (c'est-à-dire  $\pi \prec \tau$ , cf. Équation 1.1). Le signe se définit par rapport à la taille et la longueur de la composition :  $(-1)^{|\pi| - \ell(\pi)}$ .

**Exemple 2.7.1 :**

$$\mathbf{ch}(\mathbf{E}^5; 4) = 70\Lambda^{1111} - 35\Lambda^{112} - 35\Lambda^{121} + 15\Lambda^{13} - 35\Lambda^{211} + 15\Lambda^{22} + 15\Lambda^{31} - 5\Lambda^4 \quad (2.6'')$$

Voici quelques-uns des 15 chemins associés au monôme  $\Lambda^{13}$  :



## 2.8. Formule de $q$ -énumération

Une statistique intéressante sur les chemins décorés  $s = (Q_i)$  est l'aire sous le chemin. Celle-ci est évidemment invariante par l'action du groupe symétrique ou de l'algèbre de Hecke à  $q = 0$ . On pose

$$\text{area}(s) = \sum_{1 \leq i \leq k} (i-1) \#Q_i.$$

Cette statistique étant invariante pour nos deux actions, elle permet de raffiner encore les caractères de Frobenius en fonctions de l'aire du chemin. On a alors les  $q$ -caractères de Frobenius suivants :

$$\begin{aligned} \mathbf{ch}_q(\mathbf{E}^k; n) &= h_n([k]_q \cdot \mathbb{X}) = \sum_{\lambda \vdash n} m_\lambda([k]_q) h_\lambda, \\ \mathbf{ch}_q(\mathbf{E}^k; n) &= \mathbf{S}_n([k]_q \cdot \mathbb{A}) = \sum_{\pi \models n} \mathbf{M}_\pi([k]_q) \mathbf{S}^\pi. \end{aligned} \quad (2.7)$$

**Remarque 2.8.1 :** La spécialisation de  $\mathbf{M}_\pi$  au  $q$ -entier  $[k]_q$  ne se résume pas au  $q$ -binomial  $\begin{bmatrix} k \\ \ell(\pi) \end{bmatrix}_q$  contrairement à la spécialisation de l'entier  $k$  qui se traduit par le binomial  $\binom{k}{\ell(\pi)}$ . En particulier car ce dernier n'encode pas les multiplicités dues à la composition  $\pi$ .

**Exemple 2.8.2 :** Les  $q$ -caractères (commutatif et non-commutatif) des séquences de 5 ensembles sur les ensembles à 4 éléments sont :

$$\begin{aligned} \text{ch}_q(\mathbf{E}^5; 4) = & (q^{10} + q^9 + q^8 + q^7 + q^6) h_{11111} + \\ & (q^{13} + 2q^{12} + 4q^{11} + 2q^{10} + 4q^9 + 4q^8 + 4q^7 + 2q^6 + 4q^5 + 2q^4 + q^3) h_{2111} + \\ & (q^{14} + q^{12} + 2q^{10} + 2q^8 + 2q^6 + q^4 + q^2) h_{222} + \\ & (q^{15} + q^{14} + 2q^{13} + q^{12} + q^{11} + 2q^{10} + 2q^9 + 2q^7 + 2q^6 + q^5 + q^4 + 2q^3 + q^2 + q) h_{311} + \\ & (q^{16} + q^{12} + q^8 + q^4 + 1) h_4 \end{aligned}$$

$$\begin{aligned} \mathbf{ch}_q(\mathbf{E}^5; 4) = & (q^{10} + q^9 + q^8 + q^7 + q^6) \mathbf{S}^{1111} + \\ & (q^{13} + q^{12} + 2q^{11} + q^{10} + 2q^9 + q^8 + q^7 + q^5) \mathbf{S}^{112} + \\ & (q^{12} + q^{11} + q^{10} + q^9 + 2q^8 + q^7 + q^6 + q^5 + q^4) \mathbf{S}^{121} + \\ & (q^{15} + q^{14} + q^{13} + q^{12} + q^{11} + q^{10} + q^9 + q^7 + q^6 + q^3) \mathbf{S}^{13} + \\ & (q^{11} + q^9 + q^8 + 2q^7 + q^6 + 2q^5 + q^4 + q^3) \mathbf{S}^{211} + \\ & (q^{14} + q^{12} + 2q^{10} + 2q^8 + 2q^6 + q^4 + q^2) \mathbf{S}^{22} + \\ & (q^{13} + q^{10} + q^9 + q^7 + q^6 + q^5 + q^4 + q^3 + q^2 + q) \mathbf{S}^{31} + \\ & (q^{16} + q^{12} + q^8 + q^4 + 1) \mathbf{S}^4 \end{aligned}$$

Le coefficient  $4q^8$  associé à  $h_{2111}$  correspond aux quatre séquences (à isomorphisme près) :

$$\begin{aligned} (\cdot \mid \alpha\beta \mid \gamma \mid \cdot \mid \tau) & \rightsquigarrow (q^1)^2 \cdot q^2 \cdot q^4 \mathbf{S}^{211}, \\ \left. \begin{aligned} (\cdot \mid \alpha \mid \beta\gamma \mid \tau \mid \cdot) \\ (\alpha \mid \cdot \mid \beta\gamma \mid \cdot \mid \tau) \end{aligned} \right\} & \rightsquigarrow (q^1 \cdot (q^2)^2 \cdot q^3 + q^0 \cdot (q^2)^2 \cdot q^4) \mathbf{S}^{121}, \\ (\alpha \mid \cdot \mid \beta \mid \gamma\tau \mid \cdot) & \rightsquigarrow q^0 \cdot q^2 \cdot (q^3)^2 \mathbf{S}^{112}. \end{aligned}$$

## 3. Fonctions de parking

Dans la littérature, on présente souvent les fonctions de parking comme des endofonctions de  $[n]$ , définies de manière concise comme des mots  $a_1 a_2 \cdots a_n$  (où  $i \mapsto a_i$ ), qui vérifient que le nombre d'images  $a_j \leq k$  est supérieur (ou égal) à  $k$ . Cette définition est trivialement incluse dans la précédente (Définition 1.5.20). Dans cette section, on va donner une autre définition équivalente en termes de séquences d'ensembles. À partir de cette définition utilisant l'espèce des séquences d'ensembles, on va pouvoir retrouver facilement les propriétés classiques sur les fonctions de parking.

### 3.1. Espèce des fonctions de parking $\mathcal{P}$

Les chemins de Dyck sont isomorphes aux types d'isomorphismes des fonctions de parking (pour l'action du groupe symétrique  $f \circ \sigma^{-1}$ ). Cette action sur les fonctions symétriques a été énoncée pour la première fois dans [Hai94] par Mark D. HAIMAN. Cette dernière définit un transport de la structure  $f$  le long de  $\sigma$ , autrement dit, on a une description des fonctions de parking en terme d'*espèce de structures* :

**Proposition 3.1.1 :** La donnée  $\mathcal{P}$  définit par :

- $\mathcal{P}[U]$  l'ensemble des fonctions de parking sur  $U$ , pour tout ensemble fini  $U$  ;
- $\mathcal{P}[\sigma] : f \mapsto f \circ \sigma^{-1}$ , pour toute bijection  $\sigma : U \rightarrow V$ ,

est une espèce.

PREUVE : La contrainte de parking (1.23) est indépendante du domaine de la fonction  $f$ , elle ne dépend que des ensembles d'images. D'où si  $f$  est une fonction de parking sur  $U$  alors  $f \circ \sigma^{-1}$  est une fonction de parking sur  $V$ , quelque soit la bijection  $\sigma : U \rightarrow V$ . De plus, par définition du transport de structures, la composition d'application est bien associative donc :

$$(f \circ \sigma^{-1}) \circ \tau^{-1} = f \circ (\sigma^{-1} \circ \tau^{-1}),$$

pour toutes bijection  $\sigma : U \rightarrow V$  et  $\tau : V \rightarrow W$ . La contrainte d'identité est bien évidemment aussi satisfaite  $f \circ Id = f$ . ■

### 3.2. Séquences d'ensembles et fonctions de parking

Une fonction  $f : U \rightarrow \mathbb{N}_{>0}$  peut se définir de manière équivalente comme la séquence de ses images réciproques  $(Q_i) := (f^{-1}(\{i\}))_{i \geq 1}$ . La contrainte de parking (1.23)

se traduit par :

$$\sum_{i=1}^k \#Q_i \geq k, \quad \text{pour tout } k \leq \#U. \quad (3.1)$$

Pour les mêmes raisons que  $f^{-1}(\{k\}) = \emptyset$  pour tout  $k > n := \#U$ , on peut se limiter aux séquences de  $n$  ensembles.

Dans la suite, on considère l'espèce  $\mathcal{P}$  des fonctions de parking comme l'espèce définie par :

- $\mathcal{P}[U]$  est l'ensemble des séquences  $(Q_i)$  de  $n$  ensembles qui vérifient (3.1), pour tout ensemble fini  $U$  de cardinalité  $n$ ,
- $\mathcal{P}[\sigma]((Q_i)) := (\sigma(Q_i))$ , pour toute bijection  $\sigma : U \rightarrow V$ .

On a alors que les  $\mathcal{P}$ -structures sur  $U$  sont incluses dans les  $\mathbf{E}^n$ -structures sur  $U$  (avec  $n := \#U$ ). En effet,

**Proposition 3.2.1 :** *Le transport d'une fonction de parking le long d'une bijection  $\sigma$  coïncide avec le transport des séquences d'ensembles.*

Autrement dit, l'espèce des fonctions de parking et l'espèce des séquences d'ensembles vérifiant la contrainte de parking sont isomorphes.

PREUVE : Soit  $f$  une fonction de parking sur  $U$  et  $(Q_i)$  sa version équivalente en termes de séquence d'ensembles :

$$Q_i := f^{-1}(\{i\}) \quad \text{pour tout } 1 \leq i \leq \#U.$$

On sait que  $(f \circ \sigma^{-1})^{-1} = \sigma \circ f^{-1}$  donc  $\sigma \cdot (f^{-1}(\{i\}))_i = (\sigma \circ f^{-1}(\{i\}))_i$ , ce qui correspond bien à l'action du groupe symétrique sur les séquences d'ensembles. ■

**Remarque 3.2.2 :** Pour tout  $n \geq 0$ , l'espèce  $\mathcal{P}_n$ , définie par la restriction canonique de  $\mathcal{P}$  aux ensembles de cardinalité  $n$  (cf. Définition 1.5.14), est une sous-espèce de  $\mathbf{E}^n$  l'espèce des séquences de  $n$  ensembles. Plus précisément c'est une sous-espèce de la restriction canonique  $(\mathbf{E}^n)_n$  des séquences de  $n$  ensembles sur les ensembles à  $n$  éléments.

### 3.3. $\mathcal{P}$ -structures

Dans cette sous-section, nous faisons un petit récapitulatif des différentes structures de données équivalentes de fonctions de parking. Dans la précédente section, on a représenté les séquences d'ensembles par des chemins décorés. On vient de voir que les fonctions de parking peuvent également être définies comme des séquences d'ensembles particulières satisfaisant la contrainte de parking (3.1). Après avoir rappelé les différentes structures de données : *fonctions entières*, *séquences d'ensembles* (ou comme des *mots* quand  $U = [n]$ ) ; nous allons rappeler que les chemins décorés obtenus à partir de ces séquences de parking sont exactement les *chemins de Dyck décorés*.

### 3.3.1. Fonctions entières

Les fonctions entières sont normalement utilisées dans pour la définition usuelle des fonctions de parking (Définition 1.5.20).

**Exemple 3.3.1:** Les seize fonctions de parking sur  $U = \{\alpha, \beta, \gamma\}$  définies comme des fonctions entières :

$$\begin{array}{l|l} \alpha, \beta, \gamma \mapsto 1; & \beta \mapsto 1, \alpha, \gamma \mapsto 2; \\ \alpha, \beta \mapsto 1, \gamma \mapsto 2; & \gamma \mapsto 1, \alpha, \beta \mapsto 2; \\ \alpha, \beta \mapsto 1, \gamma \mapsto 3; & \alpha \mapsto 1, \beta \mapsto 2, \gamma \mapsto 3; \\ \alpha, \gamma \mapsto 1, \beta \mapsto 2; & \alpha \mapsto 1, \gamma \mapsto 2, \beta \mapsto 3; \\ \alpha, \gamma \mapsto 1, \beta \mapsto 3; & \beta \mapsto 1, \alpha \mapsto 2, \gamma \mapsto 3; \\ \beta, \gamma \mapsto 1, \alpha \mapsto 2; & \beta \mapsto 1, \gamma \mapsto 2, \alpha \mapsto 3; \\ \beta, \gamma \mapsto 1, \alpha \mapsto 3; & \gamma \mapsto 1, \alpha \mapsto 2, \beta \mapsto 3; \\ \alpha \mapsto 1, \beta, \gamma \mapsto 2; & \gamma \mapsto 1, \beta \mapsto 2, \gamma \mapsto 3. \end{array}$$

### 3.3.2. Séquences d'ensembles

La définition en fonctions entières (Définition 1.5.20) est en bijection avec la définition en séquences d'ensembles (3.1) par :

$$f \longleftrightarrow (f^{-1}(\{k\}))_{1 \leq k \leq \#U}$$

pour tout ensemble fini  $U$ .

**Exemple 3.3.2:** Les  $\mathcal{P}$ -structures sur  $U = \{\alpha, \beta, \gamma\}$  définies comme des séquences de 3 ensembles :

$$\begin{array}{c|c|c|c} (\alpha\beta\gamma \mid \cdot \mid \cdot) & (\beta \mid \alpha\gamma \mid \cdot) & (\alpha\beta \mid \gamma \mid \cdot) & (\gamma \mid \alpha\beta \mid \cdot) \\ (\alpha\beta \mid \cdot \mid \gamma) & (\alpha \mid \beta \mid \gamma) & (\alpha\gamma \mid \beta \mid \cdot) & (\alpha \mid \gamma \mid \beta) \\ (\alpha\gamma \mid \cdot \mid \beta) & (\beta \mid \alpha \mid \gamma) & (\beta\gamma \mid \alpha \mid \cdot) & (\beta \mid \gamma \mid \alpha) \\ (\beta\gamma \mid \cdot \mid \alpha) & (\gamma \mid \alpha \mid \beta) & (\alpha \mid \beta\gamma \mid \cdot) & (\gamma \mid \beta \mid \gamma) \end{array}$$

### 3.3.3. Mots

Dans le cas où  $U$  est l'ensemble  $[n]$ , la définition en fonctions entières ou en séquences d'ensembles est en bijection avec l'ensemble des mots  $w = a_1 \cdots a_n$  tels que le trié  $w^\uparrow = a_1^\uparrow \cdots a_n^\uparrow$  vérifie :

$$a_k \leq k \quad \text{pour tout } 1 \leq k \leq n. \quad (3.2)$$

La bijection explicite est décrite par :

$$f \longleftrightarrow f(1)f(2) \cdots f(n).$$

**Remarque 3.3.3 :** Le transport de structure des fonctions de parking,  $f \circ \sigma^{-1}$ , se traduit par l'action à droite sur les mots,  $w \cdot \sigma$  (1.6), pour  $\sigma : [n] \rightarrow [n]$  une permutation de  $n$ .

**Exemple 3.3.4 :** Les  $\mathcal{P}$ -structures sur les ensembles  $\emptyset$ ,  $[1]$ ,  $[2]$  et  $[3]$  définies comme des mots :

$$\begin{aligned} \mathcal{P}[\emptyset] &= \{ \epsilon \}, & \mathcal{P}[1] &= \{ 1 \}, & \mathcal{P}[2] &= \{ 11, 12, 21 \}, \\ \mathcal{P}[3] &= \left\{ \begin{array}{l} 111, 112, 121, 211, 113, 131, 311, 122, \\ 212, 221, 123, 132, 312, 213, 231, 321 \end{array} \right\}. \end{aligned}$$

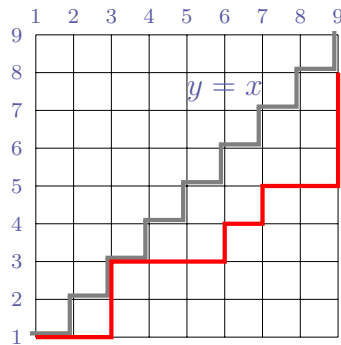
Ces trois structures de données ont chacune leurs avantages. Dans la suite de cette partie, on utilisera principalement la description en séquences d'ensembles. Cette description est celle utilisée dans les chapitres sur les fonctions de parking généralisées et les automates acycliques mais, suivant le contexte, il va être préférable d'utiliser la description en mots pour certaines preuves dans la partie sur les algèbres de Hopf de fonctions de parking généralisées. On peut encore considérer une quatrième structure de données ayant comme avantage de traduire de façon visuelle la contrainte de parking (1.23) ainsi que les types d'isomorphismes :

### 3.3.4. Chemins de Dyck décorés

Un *chemin de Dyck* de taille  $n$  est un chemin escalier composé marche et contre-marche de l'origine  $(1, 1)$  jusqu'à  $(n + 1, n)$  sous la courbe discrète  $y = x$ .

**Remarque 3.3.5 :** Encore une fois, on pourrait faire débiter nos chemins de l'origine jusqu'à  $(n, n - 1)$  mais dans la suite ce détail aura son importance.

**Exemple 3.3.6 :** Voici un chemin de Dyck de taille 8 :

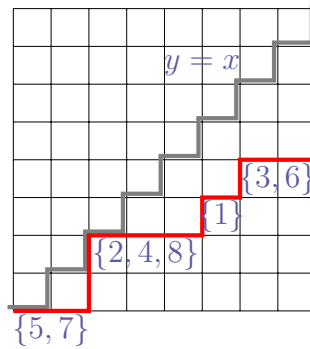


Ces chemins sont bien connus et étudiés pour de nombreuses raisons. On peut notamment noter que les chemins de Dyck  $\mathcal{D}$  ont pour série, la série génératrice des nombres de Catalan :

$$\mathcal{D}(t) = \frac{1 - \sqrt{1 - 4t}}{2t} = \sum_{n \geq 0} C_n t^n = 1 + t + 2t^2 + 5t^3 + 14t^4 + 42t^5 + \dots \quad [\text{A000108}]$$

Un *chemin de Dyck décoré* sur  $U$  est un chemin décoré défini sous la courbe discrète identité.

**Exemple 3.3.7 :** Voici un chemin de Dyck décoré sur [8] :



Cette définition des chemins de Dyck décorés est équivalente à la définition des fonctions de parking en termes de chemins décorés :

**Proposition 3.3.8 :** Les chemins de Dyck décorés sont en bijection avec les fonctions de parking.

La bijection est immédiate, les structures de données sont identiques. En effet, les chemins de Dyck décorés sont avant tout des chemins décorés. La preuve consiste alors à justifier que la contrainte d'être un chemin sous la courbe discrète  $y = x$  est équivalente à la contrainte de parking (3.1).

PREUVE: Par définition, la longueur d'une marche coïncide avec le cardinal de l'ensemble associé, et la hauteur d'une marche coïncide avec l'indice de l'ensemble qui la décore. Quant à la contrainte de parking (3.1), elle impose donc que le chemin décoré d'une fonction de parking soit un chemin de Dyck décoré et réciproquement. ■

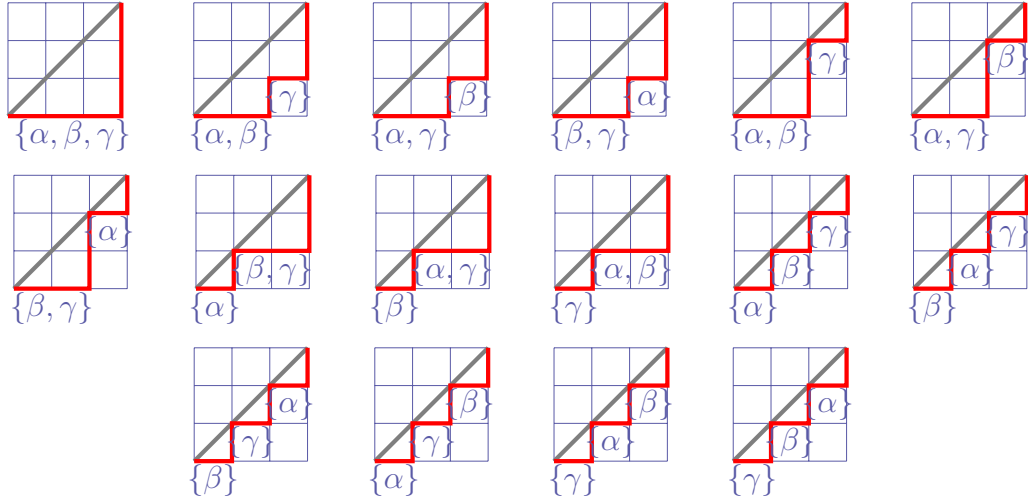
Cette nouvelle structure de donnée (les chemins de Dyck décorés) permet de déduire facilement que les types d'isomorphismes des fonctions de parking sont en bijection avec les chemins de Dyck (usuels). Ce point de vue est notamment utilisé par Nicholas A. LOEHR dans [Loe05] pour prouver les coefficients de la série de Hilbert des polynômes harmoniques diagonaux  $H_n(q, t)$ . Cette série s'exprime comme une somme de statistiques sur l'ensemble des fonctions de parking :

$$H_n(q, t) = \sum_{f \in \mathcal{P}[n]} q^{\text{area}(f)} t^{\text{dinv}(f)},$$

où  $\text{dinv}$  est une autre statistique (plus compliquée qui n'est pas traitée ici).

C'est aussi en considérant cette structure dans la prochaine section sur les fonctions de parking généralisées qu'on voit apparaître une définition récursive.

**Exemple 3.3.9 :** Les  $\mathcal{P}$ -structures sur  $\{\alpha, \beta, \gamma\}$  définies comme des chemins décorés :



### 3.4. Série génératrice

La série génératrice des fonctions de parking est :

$$\mathcal{P}(t) = \sum_{n \geq 0} (n+1)^{n-1} \frac{t^n}{n!} = 1 + t + 3\frac{t^2}{2} + 16\frac{t^3}{3!} + 1296\frac{t^4}{4!} + 262144\frac{t^5}{5!} + \dots \quad [\text{A000272}]$$

Il existe de différentes preuves de résultat, l'une des plus élégante est celle de Henry O. POLLAK (discutée dans [Rio69 ; FR74]) :

PREUVE : Soit  $U$  un ensemble fini de cardinalité  $n$ . On considère l'ensemble  $C_U$  de toutes les fonctions  $f : U \rightarrow \{0, \dots, n\}$ . On sait que

$$\#C_U = (n+1)^n.$$

On reprend l'idée que  $u$  correspond à une voiture se déplaçant dans une rue à sens unique. On suppose (cette fois-ci) qu'il s'agit d'une rue circulaire contenant  $n+1$  places de parking numérotées de  $0$  à  $n$  dans le sens de la circulation et les emplacements  $0$  et  $n$  sont adjacents et ferment le cercle. Chaque voiture se gare soit à son emplacement préféré  $f(u)$  soit sur le premier emplacement disponible  $f(u) + j$  (avec  $j$  le plus petit possible). Dans le cas d'une rue circulaire, chaque voiture trouve nécessairement une place et il existe toujours un unique emplacement libre, quelque soit  $f \in C_U$ .

Par ailleurs,  $f \in C_U$  est une fonction de parking sur  $U$  si et seulement si  $f^{-1}(\{0\})$  est vide. Ceci découle de la contrainte de parking (1.23) :

$$\{f(u) > n \mid u \in U\} = \emptyset.$$

Soit  $\sim$  la relation d'équivalence sur  $C_U$  définie par

$$f \sim g \quad \text{si et seulement si} \quad \exists j \geq 0, \forall u \in U, \quad f(u) = g(u) + j, \quad \text{mod } (n+1).$$



Ses classes d'équivalences contiennent toutes exactement  $n+1$  fonctions. Dans chaque classe d'équivalence  $[f]$ , il existe exactement une et une seule fonction  $g \in [f]$  qui est une fonction de parking sur  $U$ . On déduit donc

$$\#\mathcal{P}[U] = \frac{\#C_U}{n+1} = (n+1)^{n-1}. \quad \blacksquare$$

Dans le prochain chapitre, on donne d'autres preuves de ce résultat en appliquant une bijection de structures (parmi les multiples possibles) avec les forêts d'arbres enracinés  $\mathcal{F}$ .

### 3.5. Série indicatrice de cycles

L'approche de Henry O. POLLAK couplée au point de vue de l'espèce des séquences d'ensembles permet de donner une formule élégante de la série indicatrice de cycles :

$$\begin{aligned} \text{ch}(\mathcal{P}; n) &= \frac{1}{n+1} \text{ch}(\mathbb{E}^{n+1}; n), \\ &= \frac{1}{n+1} [t^n] \sigma_t(\mathbb{X})^{n+1}, \\ &= \frac{1}{n+1} h_n((n+1) \cdot \mathbb{X}), \end{aligned}$$

et de même en version non-commutative :

$$\mathbf{ch}(\mathcal{P}; n) = \frac{1}{n+1} \mathbf{S}_n(n+1 \cdot \mathbb{A}).$$

On peut trouver de nombreuses autres formulations du caractère de Frobenius (commutatif) dans [Sta97, Proposition 2.2] notamment dans les bases des fonctions de Schur, des fonctions monomiales et des sommes de puissances. (Elles découlent simplement de la traduction de  $h_n((n+1) \cdot \mathbb{X})$  dans les autres bases.)

De ces caractères, on retrouve le nombre de fonctions de parking  $(n+1)^{n-1}$  ainsi que les nombres de Catalan associés aux coefficients de  $\mathcal{D}(t) = \widetilde{\mathcal{P}}(t)$ , à partir de l'Équation 2.4 :

$$\widetilde{\mathcal{P}}(t) = \sum_{n \geq 0} \frac{[t^n] \widetilde{\mathbb{E}^{n+1}}(t)}{n+1} t^n = \sum_{n \geq 0} \frac{\binom{2n}{n}}{n+1} t^n. \quad [\text{A000108}]$$

#### 3.5.1. $q$ -analogues

Dans [NT08], Jean-Christophe NOVELLI et Jean-Yves THIBON déduisent le  $q$ -caractère de Frobenius non-commutative des fonctions de parking sous l'action des l'algèbre de 0-Hecke dégénérée (où  $q$  encode l'aire) par l'équation :

$$\mathbf{S}^n([n]_q \cdot \mathbb{A}) = \sum_{k=0}^n \mathbf{ch}_q(\mathcal{P}; k) q^{(k+1)(n-k)} \mathbf{S}^{n-k}([n-k-1]_q \cdot \mathbb{A}). \quad (3.3)$$

Cette relation découle d'une propriété déjà exposée dans [KW66, § 3], on peut décomposer (de manière unique) toute séquence  $(Q_i)$  de  $\mathbf{E}^{n+r}$  (avec  $r \geq 1$ ) sur un ensemble fini  $U$  (à  $n$  éléments) comme le plus grand préfixe possible  $(X_j)$  qui définit une fonction de parking sur un sous-ensemble  $S$  de  $U$  (c'est-à-dire le plus grand préfixe qui satisfait (3.1)), suivi du suffixe  $(Y_j)$  (qui est une séquence d'ensembles sur  $U \setminus S$ ) où le premier ensemble de la séquence est nécessairement vide :

**Proposition 3.5.1 :**

$$\mathbf{E}^{n+r}[U] = \sum_{\substack{S \sqcup T = U \\ s := \#S}} \mathcal{P}[S] \cdot (\mathbf{E}_0 \cdot \mathbf{E}^{n+r-s-1})[T].$$

**Exemple 3.5.2 :** Voici le découpage de quelques-unes des soixante-quatre  $\mathbf{E}^{3+1}$ -structures sur  $U := \{a, b, c\}$  où le préfixe en bleu représente la plus grande sous-séquence qui est une fonction de parking :

$$\dots, ()(\cdot | \cdot | b | ac), ()(\cdot | \cdot | ac | b), ()(\cdot | \cdot | ab | c), ()(\cdot | \cdot | bc | a), \dots$$

se découpent en  $\mathcal{P}$ -structures sur l'ensemble vide concaténées à  $\mathbf{E}^4$ -structures où le premier ensemble est vide ;

$$(a)(\cdot | \cdot | bc), (a)(\cdot | c | b), (a)(\cdot | b | c), (a)(\cdot | bc | \cdot)$$

se découpent en  $\mathcal{P}$ -structures sur  $\{a\}$  et en séquence de  $\mathbf{E}^3$  sur  $\{b, c\}$  où le premier ensemble est vide ; etc. jusqu'à

$$\dots, (c | ab | \cdot)(\cdot), (b | a | c)(\cdot), (b | c | a)(\cdot), (b | ac | \cdot)(\cdot), \dots$$

qui se découpent en fonctions de parking sur  $\{a, b, c\}$  et en séquences d'un ensemble vide.

La relation du  $q$ -caractère (3.3) se déduit alors à partir du  $q$ -caractère des séquences d'ensembles (2.7). L'exposant  $(k+1)(n-k)$  de  $q$  correspond simplement à normaliser l'aire des chemins. Le préfixe de parking  $(X_j)$  est une séquence de  $k$  ensembles sur l'ensemble  $S$  à  $k$  éléments, c'est-à-dire un chemin décoré de longueur et hauteur  $k$  ; d'où le suffixe  $(Y_j)$  est un suffixe du chemin décoré de longueur  $n-k$  débutant au moins à hauteur  $k+1$ .

**Exemple 3.5.3 :**

$$\begin{aligned} \mathbf{ch}(\mathcal{P}; 4) = & q^6 \mathbf{S}^{1111} + q^5 \mathbf{S}^{112} + (q^5 + q^4) \mathbf{S}^{121} + q^3 \mathbf{S}^{13} + (q^5 + q^4 + q^3) \mathbf{S}^{211} + \\ & (q^4 + q^2) \mathbf{S}^{22} + (q^3 + q^2 + q) \mathbf{S}^{31} + \mathbf{S}^4 \end{aligned}$$

Chaque coefficient  $\alpha$  associée à un terme  $q^n \mathbf{S}^\pi$  de ce  $q$ -caractère encode le nombre de chemin de Dyck ou fonctions de parking (à isomorphisme près)  $d$ , telles que l'aire est  $\mathbf{aread} = n$  et l'évaluation tassée est  $\mathbf{pEv}(d) = \pi$  :

$$\mathbf{ch}_q(\mathcal{P}; n) = \sum_{d \in \mathcal{D}_n} q^{\mathbf{aread}(d)} \mathbf{S}^{\mathbf{pEv}(d)},$$

où  $\mathcal{D}_n$  est l'ensemble des chemins de Dyck  $d$  de longueur  $n$ .

## 4. Fonctions de parking généralisées

Après avoir introduit les séquences d'ensembles et les fonctions de parking usuelles (en terme de séquences d'ensembles), on introduit les *fonctions de parking généralisées*. Dans [PS02], Richard P. STANLEY et Jim PITMAN introduisent ces structures de *fonctions de parking généralisées*. Soit  $\chi : \mathbb{N}_{>0} \rightarrow \mathbb{N}$  une fonction croissante. Une  $\chi$ -fonction de parking  $(Q_i)$  sur  $U$  (de cardinalité  $n$ ) est une séquence de  $\chi(n)$  sous-ensembles disjoints de  $U$  telle que :

$$\sum_{i=1}^{\chi(k)} \#Q_i \geq k \quad \text{pour tout } 1 \leq k \leq n. \quad (4.1)$$

On note  $\mathcal{P}_\chi$  ou  $\mathcal{P}(\chi)$  l'espèce des  $\chi$ -fonctions de parking (où le transport de structure est défini comme pour les fonctions de parking usuelles :  $\sigma \cdot (Q_i)_i = (\sigma(Q_i))_i$ ). Avec un léger abus, on notera souvent directement l'image de  $\chi$  par  $m$  au lieu de la définir formellement (cf. l'exemple suivant). Dans la suite, on fera la supposition que  $\chi$  est une endofonction croissante de  $\mathbb{N}$  avec  $\chi(0) = 0$  par convention.

**Exemple 4.0.4 :** L'ensemble des onze  $\lceil \frac{m}{2} \rceil$ -fonctions de parking (autrement dit, les  $\mathcal{P}(\chi)$ -structures pour  $\chi$  la fonction associant à  $m \in \mathbb{N}_{>0}$  la partie entière supérieure de  $m/2$ ), sur l'ensemble [4] :

$$(1234 \mid \cdot), (123 \mid 4), (124 \mid 3), (134 \mid 2), (234 \mid 1), \\ (12 \mid 34), (13 \mid 24), (14 \mid 23), (23 \mid 14), (24 \mid 13), (34 \mid 12).$$

**Remarque 4.0.5 :** Les fonctions de parking usuelles sont les fonctions de parking généralisées où  $\chi$  est l'identité. Autrement dit, on a  $\mathcal{P} = \mathcal{P}(m)$ .

Dans [PS02], les auteurs déduisent une élégante formule pour énumérer les  $\chi$ -fonctions de parking à partir des fonctions de parking usuelles :

$$p(\chi; n) = \sum_{f \in \mathcal{P}[n]} \prod_{i=1}^n \left( \chi(f(i)) - \chi(f(i) - 1) \right), \quad [\text{PS02, Théorème 11}]$$

où  $p(\chi; n)$  est le nombre de  $\chi$ -fonctions de parking sur un ensemble à  $n$  éléments.

Cette formule se traduit comme le cas discret d'un calcul de volume de polytopes de dimension  $n$  [PS02, Théorème 1].

Cette généralisation, dans le cas des fonctions  $\chi$  rationnelles, a été aussi étudiée dans *Enumerative combinatorics, Volume II* (Exercice 5.49). L'auteur donne une jolie formule d'énumération des fonctions de parking rationnelles :

$$p(a(m-1) + b; n) = b(an + b)^{n-1},$$

pour tous  $a, b$  entiers strictement positifs. Cette formule met en évidence des liens avec ces fonctions de parking avec des structures arborescentes (cf. § 5.2).

Des cas similaires, où  $\chi$  est « presque » rationnelle, ont été étudiés par C. H. YAN dans [Yan00 ; Yan01]. L'auteure a considéré le cas où la suite  $\chi$  est définie comme la somme partielle de vecteurs finis, de la forme  $(a, b, \dots, b, c, \dots, c)$  (pour un certain nombre d'entiers  $b$  et un certain nombre d'entiers  $c$  fixés), étendus à l'infini par des 0.

Ce même auteur s'est aussi intéressé à l'énumération des  $\chi$ -fonctions de parking en utilisant une base de polynômes de Gončarov avec Joseph P.S. KUNG [KY03] puis avec Xinyu SUN [KSY06]. Ils obtiennent la formule déterminantale :

$$D(\chi, t; n) = n! \begin{vmatrix} 1 & \chi(1) & \frac{\chi(1)^2}{2!} & \frac{\chi(1)^3}{3!} & \dots & \frac{\chi(1)^{n-1}}{(n-1)!} & \frac{\chi(1)^n}{n!} \\ 0 & 1 & \chi(2) & \frac{\chi(2)^2}{2!} & \dots & \frac{\chi(2)^{n-2}}{(n-2)!} & \frac{\chi(2)^{n-1}}{(n-1)!} \\ 0 & 0 & 1 & \chi(3) & \dots & \frac{\chi(3)^{n-3}}{(n-3)!} & \frac{\chi(3)^{n-2}}{(n-2)!} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 1 & \chi(n) \\ 1 & t & \frac{t^2}{2!} & \frac{t^3}{3!} & \dots & \frac{t^{n-1}}{(n-1)!} & \frac{t^n}{n!} \end{vmatrix}$$

pour laquelle on a l'égalité suivante :

$$p(\chi; n) = (-1)^n D(\chi, 0; n).$$

Dans cette section, on s'intéresse à définir l'espèce des  $\chi$ -fonctions de parking par l'intermédiaire d'une grammaire récursive [PV15]. Une fonction de parking  $(Q_i)$  peut se découper comme une séquence de  $\chi(1)$  ensembles sans aucune contrainte suivie d'une  $\psi$ -fonction de parking avec  $\psi$  un décalage particulier de  $\chi$ . L'intérêt de ces définitions constructives est d'obtenir automatiquement de nouvelles formules d'énumérations. Nous décrirons notamment le caractère de Frobenius des fonctions de parking généralisées sous l'action de l'algèbre de Hecke à  $q = 0$  dans différentes bases des fonctions symétriques non-commutatives. Ce caractère va se révéler extrêmement riche et expressif pour l'énumération des automates acycliques (Chapitre Chapitre 8), via une bijection bien choisie avec une sous-famille de  $\chi$ -fonctions de parking.

## 4.1. Définition récursive

Soit  $(Q_i)$  une  $\chi$ -fonction de parking sur un ensemble  $U$  à  $n$  éléments.

### 4.1.1. Intuition sur les chemins décorés

On rappelle que les fonctions de parking usuelles sont en bijection avec les chemins de Dyck décorés, autrement dit avec les chemins décorés sous la courbe discrète identité  $\chi : m \mapsto m$ . Une fonction de parking généralisée peut se décrire de la même manière par un chemin décoré vérifiant que le chemin est sous la courbe discrète  $\chi$ .

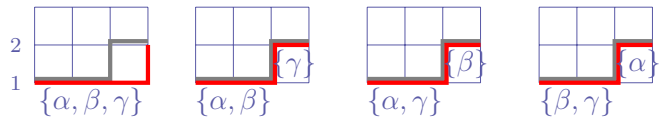
**Proposition 4.1.1 :** Les  $\chi$ -fonctions de parking sont en bijection avec les chemins décorés sous la courbe discrète  $\chi$ .

PREUVE : Analogue à la preuve Proposition 3.3.8. ■

**Exemple 4.1.2 :** Soit  $\chi$  défini par la séquence 1, 1, 2, 2, 3, 3,  $\dots$ . Les quatre  $\chi$ -fonctions de parking sur  $\{\alpha, \beta, \gamma\}$  :

$$(\alpha\beta\gamma \mid \cdot), (\alpha\beta \mid \gamma), (\alpha\gamma \mid \beta), (\beta\gamma \mid \alpha)$$

correspondent respectivement aux quatre chemins décorés (en rouge) sous la courbe discrète  $\chi$  (en gris) :

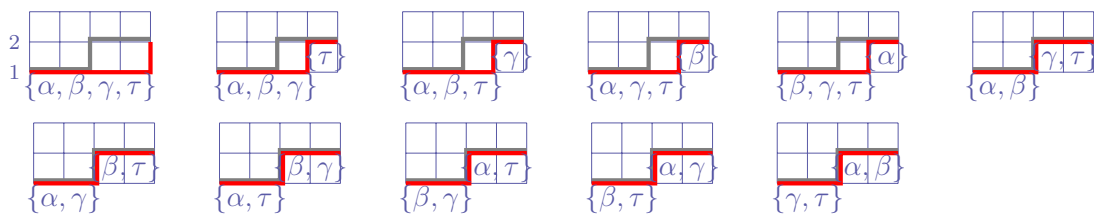


et, les onze  $\chi$ -fonctions de parking sur  $\{\alpha, \beta, \gamma, \tau\}$  :

$$(\alpha\beta\gamma\tau \mid \cdot), (\alpha\beta\gamma \mid \tau), (\alpha\beta\tau \mid \gamma), (\alpha\gamma\tau \mid \beta), (\beta\gamma\tau \mid \alpha), (\alpha\beta \mid \gamma\tau),$$

$$(\alpha\gamma \mid \beta\tau), (\alpha\tau \mid \beta\gamma), (\beta\gamma \mid \alpha\tau), (\beta\tau \mid \alpha\gamma), (\gamma\tau \mid \alpha\beta)$$

correspondent respectivement aux chemins :



La définition récursive des  $\chi$ -fonctions de parking apparaît naturellement lorsqu'on considère ces séquences comme des chemins décorés. Considérons le chemin décoré  $C$  sous la courbe discrète  $\chi$  (cf. Figure 4.1). Cette  $\chi$ -fonction de parking se factorise en une première partie  $S$  sous  $\chi(1)$  suivie d'une  $\psi$ -fonction de parking  $F$ . La contrainte de parking (4.1) n'impose rien sur les  $\chi(1)$  ensembles de  $S$ . Autrement dit  $S$  est une séquence quelconque de  $\mathbf{E}^{\chi(1)}$  sur un ensemble à  $k$  éléments. Par ailleurs, la suite de cette séquence,  $F$  est un chemin sous la courbe discrète définie par la suite  $\chi(k+1), \chi(k+2), \chi(k+3), \dots$ . De plus, ce chemin est toujours au-dessus de  $\chi(1)$ . On peut donc décrire  $F$  comme une fonction de parking généralisée associée à une certaine fonction  $\psi_k$  définie comme une translation du plan en fonction de  $\chi$  et  $k$ .

**Définition 4.1.3 :** On appelle  $k$ -translation de  $\chi$ , notée  $\psi_k$ , l'application définit par :

$$\psi_k : m \mapsto \chi(m+k) - \chi(1). \quad (4.2)$$

Cette définition est utilisée fréquemment et implicitement dans toute la suite de cette partie.

**Remarque 4.1.4 :** Afin d'être homogène avec la convention  $\chi(0) = 0$  (cf. le paragraphe suivant l'Équation 4.1), on pose  $\psi_k(0) = 0$ .

**Exemple 4.1.5 :** Soit  $\chi$  la suite de Fibonacci (avec  $\chi(1) = \chi(2) = 1$ ). La 3-translation de  $\chi$  est la suite 2, 4, 7, 12, 20, 33, 54, 88, 143,  $\dots$ .

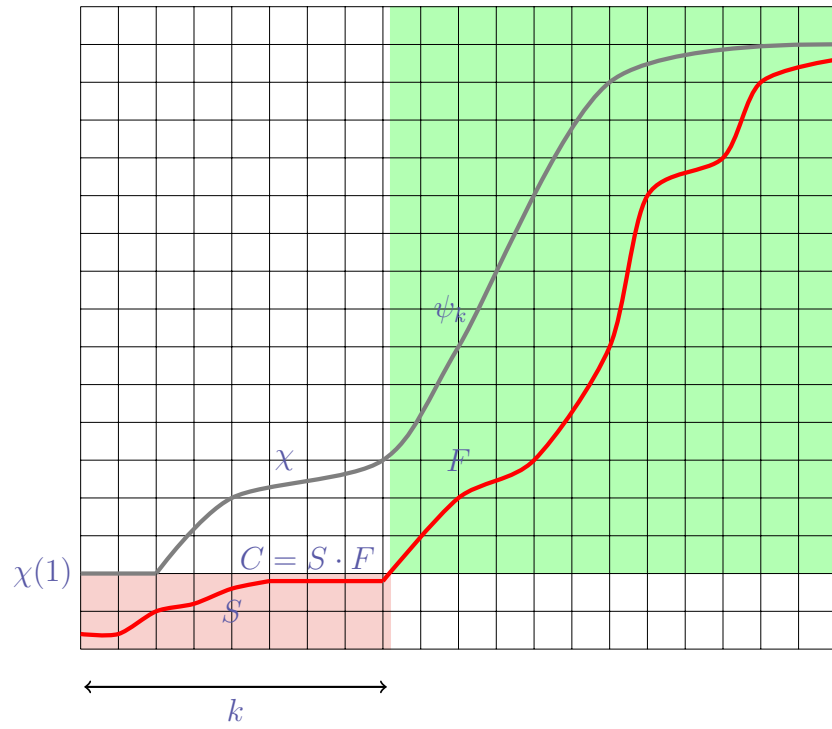


FIGURE 4.1. – Soit  $C$  une  $\chi$ -fonction de parking représentée par le chemin décoré (en rouge) sous la courbe discrète  $\chi$  (en gris). Le chemin  $C$  se décompose en chemin  $S$  de  $k$  marches sous  $\chi(1)$  (la partie orangée) suivi d'un chemin décoré  $F$  (la partie verdie). Ce chemin  $F$  est une  $\psi_k$ -fonction de parking avec  $\psi_k : m \mapsto \chi(m + k) - \chi(1)$ .

### 4.1.2. Grammaire réursive

Soit  $\chi : \mathbb{N}_{>0} \rightarrow \mathbb{N}$  croissante. On pose  $\mathbf{F}$  l'espèce définie récursivement par :

$$\mathbf{F}(\chi) := (\mathbf{E}^{\chi(1)})_0 + \sum_{n \geq 1} (\mathbf{E}^{\chi(1)})_n \cdot \mathbf{F}(\psi_n). \quad (4.3)$$

Par définition pour  $U = \emptyset$ , l'ensemble de structures est bien défini :  $\mathbf{F}(\chi)[\emptyset] = \mathbf{E}^{\chi(1)}[\emptyset]$ . La famille d'espèces  $((\mathbf{E}^{\chi(1)})_n \cdot \mathbf{F}(\psi_n))_{n \geq 1}$  est sommable. En effet pour tout ensemble fini  $U$  à  $n$  éléments (avec  $n \geq 1$ ), il existe exactement  $n$  indices  $(1, 2, \dots, n)$  tels que l'ensemble de structures soit non-vide. De plus, on remarque aisément que pour tout  $n$ ,  $\mathbf{F}(\chi)$  se développe comme un produit fini de séquences d'ensembles. Cette grammaire définit donc bien une espèce.

**Lemme 4.1.6 :** *L'espèce  $\mathbf{F}(\chi)$  est isomorphe à l'espèce des fonctions de parking généralisées  $\mathcal{P}(\chi)$ .*

PREUVE : Soit  $U$  un ensemble à  $n$  éléments et soit  $s = (Q_i)$  une  $\mathbf{F}$ -structure sur  $U$ .

**Longueur de la séquence  $(Q_i)$**  Par induction on montre aisément que la longueur d'une  $\mathbf{F}$ -structure sur  $U$  est  $\chi(n+1)$ .

**Suffixe vide** On montre également que pour tout  $i$  entre  $\chi(n) + 1$  et  $\chi(n+1)$ , les ensembles  $Q_i$  sont nécessairement vides : d'après la définition, la grammaire se termine par une séquence d'ensembles vides. Si  $U = \emptyset$ , toute la séquence est vide donc en particulier entre  $\chi(0) + 1 = 1$  et  $\chi(1)$ . Si  $U \neq \emptyset$ , on déduit la définition de  $\psi_n$  en itérant la grammaire jusqu'à cette séquence terminant la récursion, on a :

$$\begin{aligned} \psi_{n_1} : m &\mapsto \chi(m + n_1) - \chi(1), \\ \psi_{n_2} : m &\mapsto \chi(m + n_1 + n_2) - \chi(1 + n_1), \\ \psi_{n_3} : m &\mapsto \chi(m + n_1 + n_2 + n_3) - \chi(1 + n_1 + n_2), \\ &\vdots \\ \psi_{n_k} : m &\mapsto \chi(m + n) - \chi(1 + n - n_k), \end{aligned}$$

avec  $n_j \geq 1$  pour tout  $1 \leq j \leq k$ . On déduit donc que ce dernier suffixe d'ensembles vides est de longueur  $\psi_{n_k}(1) \geq \chi(n+1) - \chi(n)$ . Autrement dit, les ensembles  $Q_i$  sont vides, pour tout  $i$  entre  $\chi(n) + 1$  et  $\chi(n+1)$ .

**Contrainte de parking** Si  $U = \emptyset$  alors  $(Q_i)$  satisfait la contrainte de parking sinon par induction sur la structure de  $(Q_i)$  : d'après la définition  $(Q_i)$  se décompose comme un préfixe qui est une séquence de  $\chi(1)$  ensembles sur un sous-ensemble de  $U$  contenant  $n_1$  éléments avec  $n_1 \geq 1$ , d'où

$$\sum_{i=1}^{\chi(1)} \#Q_i = n_1 \geq 1.$$

D'après la définition encore, le suffixe  $S = (S_i)_{1 \leq i \leq \psi_{n_1}(n-n_1+1)} = (Q_i)_{\chi(1)+1 \leq i \leq \chi(n+1)}$  associé est :

- Soit une séquence de  $\chi(n+1) - \chi(1)$  ensembles vides (si  $n_1 = n$ ), et, dans ce cas, la contrainte de parking est satisfaite :

$$\underbrace{\sum_{i=1}^{\chi(1)} \#Q_i}_n + \underbrace{\sum_{i=\chi(1)+1}^{\chi(k)} \#Q_i}_0 \geq k \quad \text{pour tout } 1 \leq k \leq n.$$

- Soit une  $\psi_{n_1}$ -fonction de parking, et, on vient de montrer que

$$\sum_{i=1}^{\psi_{n_1}(1)} \#S_i \geq 1.$$

c'est-à-dire

$$\sum_{i=1}^{\chi(1+n_1)-\chi(1)} \#S_i = \sum_{i=\chi(1)+1}^{\chi(1+n_1)} \#Q_i \geq 1.$$

On a donc

$$\sum_{i=1}^{\chi(k)} \#Q_i \geq n_1 \quad \text{pour tout } 1 \leq k \leq n_1,$$

et

$$\sum_{i=1}^{\chi(n_1+1)} \#Q_i \geq n_1 + 1.$$

Par induction, on déduit alors que  $(Q_i)$  satisfait la contrainte de parking.

Les  $\chi$ -fonctions de parking ont été définies comme des séquences de  $\chi(n)$  ensembles. Sans perdre de généralité, on peut supprimer des séquences  $(Q_i) \in \mathbf{F}[U]$  le suffixe d'ensembles vides de longueur  $\chi(n+1) - \chi(n)$ , on obtient alors une séquence d'ensembles de  $\mathcal{P}_\chi[U]$ .

**Réciproquement** Si  $U = \emptyset$ , l'unique  $\chi$ -fonction de parking vide est un préfixe de l'unique séquence de  $\chi(1)$  ensembles vides. L'unicité de ces structures nous assure et définit une bijection triviale.

En reprenant l'idée de la Figure 4.1, toute  $\chi$ -fonction de parking  $(Q_i)$  (sur  $U \neq \emptyset$ ) peut se décomposer comme un préfixe :

$$(Q_1, \dots, Q_{\chi(1)}),$$



qui est une séquence de  $\mathbf{E}^{\chi(1)}$  et un suffixe :

$$(Q_{\chi(1)+1}, \dots, Q_{\chi(n)}).$$

Le préfixe est une  $\mathbf{E}^{\chi(1)}$ -structure sur  $R \subset U$ . Pour  $r = \#R$ , ce préfixe est une structure de la restriction  $(\mathbf{E}^{\chi(1)})_r$ .

Par ailleurs, le suffixe est alors une  $\psi_r$ -fonction de parking pour  $\psi_r$  la  $r$ -translation de  $\chi$ .

Par induction, on déduit que toute  $\chi$ -fonction de parking se factorise comme un préfixe des  $\mathbf{F}(\chi)$ -structures défini par la grammaire (4.3) suivit d'une séquence d'ensembles vides (définie par la partie terminale de la grammaire). On déduit donc que toute  $\chi$ -fonction de parking sur  $U$  complétée d'une séquence de  $\chi(n+1) - \chi(n)$  ensembles vides est une  $\mathbf{F}(\chi)$ -structure sur  $U$ .

**Transport de structure** Trivialement quelque soit la structure considérée, de  $\mathcal{P}(\chi)$  ou de  $\mathbf{F}(\chi)$ , le transport de structure est défini de la même manière sur les ensembles.

On a bien donc un isomorphisme entre les espèces. ■

## 4.2. Caractère de Frobenius non-commutatif

### 4.2.1. Fonctions complètes

Cette grammaire (4.3) fournit automatiquement des formules d'énumérations. En particulier, on déduit immédiatement le  $q$ -caractère de Frobenius non-commutatif :

$$\mathbf{ch}_q(\mathcal{P}_\chi; n) = \sum_{k=1}^n \mathbf{ch}_q(\mathbf{E}^{\chi(1)}; k) \cdot q^{\chi(1)(n-k)} \cdot \mathbf{ch}_q(\mathcal{P}_{\psi_k}; n-k),$$

où  $q^{\chi(1)(n-k)}$  normalise le décalage engendré par la  $k$ -translation de  $\chi$ . À partir de l'Équation (2.7), on a alors

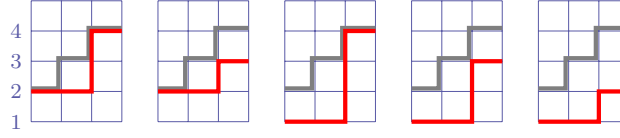
$$\mathbf{ch}_q(\mathcal{P}_\chi; n) = \sum_{k=1}^n \mathbf{S}_k \left( [\chi(1)]_q \cdot \mathbb{A} \right) \cdot q^{\chi(1)(n-k)} \cdot \mathbf{ch}_q(\mathcal{P}_{\psi_k}; n-k), \quad (4.4)$$

pour  $n \geq 1$  et  $\mathbf{ch}_q(\mathcal{P}_\chi; 0) = 1$ .

**Exemple 4.2.1 :**

$$\begin{aligned} \mathbf{ch}_q(\mathcal{P}(m+1); 3) &= (q^6 + q^5 + q^4 + q^3) \mathbf{S}^{111} + (q^5 + q^4 + q^2) \mathbf{S}^{12} \\ &\quad + (q^5 + q^4 + q^3 + q^2 + q) \mathbf{S}^{21} + (q^3 + 1) \mathbf{S}^3 \end{aligned}$$

avec les coefficients  $q^5 + q^4 + q^3 + q^2 + q$  de  $\mathbf{S}^{21}$  encodent respectivement l'aire sous les cinq chemins (en rouge) ayant un premier pas de longueur 2 et un second (et dernier) de longueur 1 :



**Remarque 4.2.2 :** L'exposant  $\chi(1)(n-k)$  de  $q$  dans la formule (4.4) permet de normaliser le membre de droite dans le produit. Les (parties de) chemins issus du terme de droite sont de longueur  $n-k$  et débutent à hauteur au moins  $\chi(1)+1$ . Cet exposant compense la  $k$ -translation de  $\chi$ .

**Remarque 4.2.3 :** La formule de ce caractère n'est pas une généralisation de la formule proposée par J.-C. NOVELLI et J.-Y. THIBON dans [NT08] (cf. Équation 3.3). La formule (4.4) s'obtient immédiatement de la définition récursive (4.3) alors que la formule (3.3) se base sur le découpage des séquences d'ensembles en un préfixe qui est une fonction de parking et un suffixe quelconque (cf. Proposition 3.5.1).

En dépliant la récurrence (4.4), on obtient alors une somme sur toutes les compositions de  $n$  :

$$\begin{aligned} \mathbf{ch}_q(\mathcal{P}_\chi; n) = \sum_{\substack{\pi \models n \\ \pi = \pi_1 \pi_2 \dots \pi_k}} \mathbf{S}_{\pi_1} \left( [\chi(1)]_q \cdot \mathbb{A} \right) q^{\chi(1)\pi_1} \mathbf{S}_{\pi_2} \left( [\chi(1 + \pi(1)) - \chi(1)]_q \cdot \mathbb{A} \right) \dots \\ q^{\chi(1 + \pi(k-1))\pi_k} \mathbf{S}_{\pi_k} \left( [\chi(1 + \pi(k-1)) - \chi(1 + \pi(k-2))]_q \cdot \mathbb{A} \right) \end{aligned}$$

Pour simplifier les énoncés, on prend comme convention que la somme partielle des parts d'une composition d'entiers est définie en  $-1$  avec  $\pi(-1) := -1$ . Avec la même volonté simplificatrice, on pose  $\Upsilon$  l'application :

$$\Upsilon(\chi; \pi, i) := \chi(1 + \pi(i-1)) - \chi(1 + \pi(i-2)) . \quad (4.5)$$

On a alors :

**Lemme 4.2.4 :**

$$\mathbf{ch}_q(\mathcal{P}_\chi; n) = \sum_{\pi \models n} \prod_{i=1}^{\ell(\pi)} q^{\chi(1 + \pi(i-1))\pi_i} \cdot \mathbf{S}_{\pi_i} \left( [\Upsilon(\chi; \pi, i)]_q \cdot \mathbb{A} \right) .$$

La spécialisation à  $q = 1$  permet de retrouver le caractère :

$$\mathbf{ch}(\mathcal{P}_\chi; n) = \sum_{\pi \models n} \prod_{i=1}^{\ell(\pi)} \mathbf{S}_{\pi_i} (\Upsilon(\chi; \pi, i) \cdot \mathbb{A}) . \quad [\text{PV15, Lemme 3.1}]$$

À partir de cette identité, on déduit relèvement non-commutatif de [KY03, Corollaire 5.6] (cf. les formules ci-après, § 4.4) faisant intervenir le produit interne (§ 1.3.2) :

**Corollaire 4.2.5 :** Si  $\chi(m) = b \cdot \psi(m)$  quelque soit  $m$  alors

$$\mathbf{ch}(\mathcal{P}_\chi; n) = \mathbf{ch}(\mathcal{P}_\psi; n) \diamond \sigma_1(b \cdot \mathbb{A}) .$$

PREUVE :

$$\mathbf{ch}(\mathcal{P}_\chi; n) = \sum_{\pi \models n} \prod_{i=1}^{\ell(\pi)} S_{\pi_i} (\Upsilon(\chi; \pi, i) \cdot \mathbb{A}) , \quad [\text{PV15, Lemme 3.1}]$$

par hypothèse, on a  $\chi(m) = b \psi(m)$ , d'où

$$\mathbf{ch}(\mathcal{P}_\chi; n) = \sum_{\pi \models n} \prod_{i=1}^{\ell(\pi)} S_{\pi_i} ((\Upsilon(\psi; \pi, i) \cdot b) \cdot \mathbb{A}) ,$$

en appliquant les propriétés des transformations d'alphabets sur les fonctions symétriques non-commutatives (cf. Remarque 4.2.6 et [KLT97]) :

$$\mathbf{ch}(\mathcal{P}_\chi; n) = \sum_{\pi \models n} \prod_{i=1}^{\ell(\pi)} S_{\pi_i} (\Upsilon(\psi; \pi, i) \cdot \mathbb{A}) \diamond \sigma_1(b \cdot \mathbb{A}) , \quad [\text{KLT97, Lemme 4.19}]$$

et enfin par linéarité du produit interne  $\diamond$  :

$$\mathbf{ch}(\mathcal{P}_\chi; n) = \left( \prod_{i=1}^{\ell(\pi)} S_{\pi_i} (\Upsilon(\psi; \pi, i) \cdot \mathbb{A}) \right) \diamond \sigma_1(b \cdot \mathbb{A}) . \quad \blacksquare$$

**Remarque 4.2.6 :** Le Lemme 4.19 de Daniel KROB, Bernard LECLERC et Jean-Yves THIBON dans [KLT97] s'énonce ainsi : soient  $\mathbb{X}$  et  $\mathbb{Y}$  deux alphabets commutatifs totalement ordonnés (à savoir  $\mathbb{X} := \Upsilon(\psi; \pi, i)$  et  $\mathbb{Y} := b$ ) et soit  $\mathbb{A}$  un alphabet non-commutatif. Pour toute fonction non-commutative  $F$ ,

$$F((\mathbb{X} \times \mathbb{Y}) \cdot \mathbb{A}) = F(\mathbb{X} \cdot \mathbb{A}) \diamond \sigma_1(\mathbb{Y} \cdot \mathbb{A}) .$$

En développant les spécialisations  $\mathbf{S}_n(k \cdot \mathbb{A})$  de  $\mathbf{ch}(\mathcal{P}_\chi; n)$ , on obtient une somme de  $\mathbf{S}^\pi$  dont les coefficients  $\alpha(\pi)$  encodent tous les chemins dont la longueur des marches (de longueur non nulle) ainsi que leur ordre d'apparition sont fixés par la valeur et l'ordre des parts de  $\pi$  (cf. Exemple 4.2.1). On peut calculer de différentes manières ces coefficients. Une première manière consiste à développer l'expression (4.4).

### 4.2.2. Développement des opérateurs d'Adams

On écrit  $\Upsilon_{\pi}^{(i)}$  pour  $\Upsilon(\chi; \pi, i)$  :

$$\mathbf{ch}_q(\mathcal{P}_{\chi}; n) = \sum_{\pi \models n} \prod_{i=1}^{\ell(\pi)} q^{\chi(1+\pi(i-1))\pi_i} \cdot \mathbf{S}^{\pi_i} \left( [\Upsilon_{\pi}^{(i)}]_q \cdot \mathbb{A} \right)$$

en développant le produit des sommes, on obtient cette somme de compositions sur les parts de  $\pi$  :

$$\begin{aligned} \mathbf{ch}_q(\mathcal{P}_{\chi}; n) &= \sum_{\pi \models n} \prod_{i=1}^{\ell(\pi)} q^{\chi(1+\pi(i-1))\pi_i} \cdot \left( \sum_{\tau \models \pi_i} \mathbf{M}_{\tau} \left( [\Upsilon_{\pi}^{(i)}]_q \right) \mathbf{S}^{\tau} \right) \\ &= \sum_{\substack{\pi \models n \\ \pi = \pi_1 \pi_2 \dots \pi_k}} \sum_{\substack{\tau_1 \models \pi_1 \\ \tau_2 \models \pi_2 \\ \vdots \\ \tau_k \models \pi_k}} \mathbf{M}_{\tau_1} \left( [\Upsilon_{\pi}^{(1)}]_q \right) \cdot q^{\chi(1)\pi_2} \dots q^{\chi(1+\pi(k-1))\pi_k} \cdot \mathbf{M}_{\tau_k} \left( [\Upsilon_{\pi}^{(k)}]_q \right) \mathbf{S}^{\tau_1 \tau_2 \dots \tau_k} \end{aligned}$$

Cette double somme encode toutes les manières de mettre en facteur la séquence des parts des compositions  $\pi \models n$ . Toutes ces manières s'expriment comme une somme sur toutes les compositions  $\tau \models \ell(\pi)$  :

$$\mathbf{ch}_q(\mathcal{P}_{\chi}; n) = \sum_{\pi \models n} \sum_{\substack{\tau \models \ell(\pi) \\ \tau = \tau_1 \tau_2 \dots \tau_k}} \mathbf{M}_{\pi_1 \dots \pi_{\tau(1)}} \left( [\Upsilon_{\gamma}^{(1)}]_q \right) q^{\chi(1)\gamma_2} \dots q^{\chi(1+\gamma(k-1))\gamma_k} \mathbf{M}_{\pi_{\tau(k-1)+1} \dots \pi_{\tau(k)}} \left( [\Upsilon_{\gamma}^{(k)}]_q \right) \mathbf{S}^{\pi}$$

où  $\gamma := \pi(\tau(1)) (\pi(\tau(2)) - \pi(\tau(1))) \dots (\pi(\tau(k)) - \pi(\tau(k-1)))$ . On pose  $\Psi$  l'application (généralisant  $\Upsilon$  (4.5)) :

$$\Psi(\chi; \pi, \tau, i) := \chi(1 + \pi(\tau(i-1))) - \chi(1 + \pi(\tau(i-2))), \quad (4.6)$$

et,  $\phi$  l'application qui retourne une composition définie comme un facteur de la séquence des parts de la composition  $\pi \models n$  en fonction d'une autre composition  $\tau \models \ell(\pi)$  et d'un entier  $1 \leq i \leq \ell(\tau)$  :

$$\phi(\pi, \tau, i) := \pi_{\tau(i-1)+1} \pi_{\tau(i-1)+2} \dots \pi_{\tau(i)}.$$

On a alors :

**Théorème 4.2.7 :**

$$\mathbf{ch}_q(\mathcal{P}_{\chi}; n) = \sum_{\pi \models n} \alpha_q(\pi) \mathbf{S}^{\pi}$$

avec

$$\alpha_q(\pi) = \sum_{\tau \models \ell(\pi)} \prod_{i=1}^{\ell(\tau)} q^{\chi(1+\pi(\tau(i-1)))[\pi(\tau(i)) - \pi(\tau(i-1))]} \mathbf{M}_{\phi(\pi, \tau, i)} \left( [\Psi(\chi; \pi, \tau, i)]_q \right).$$

**Exemple 4.2.8 :**

$$\begin{aligned} \mathbf{ch}(\mathcal{P}(2m+1); 3) = & (q^{12} + 3q^{11} + 4q^{10} + 5q^9 + 4q^8 + 4q^7 + 3q^6 + 2q^5 + q^4 + q^3) \mathbf{S}^{111} + \\ & (q^{10} + q^9 + 2q^8 + q^7 + q^6 + q^5 + q^4 + q^2) \mathbf{S}^{12} + (q^6 + q^3 + 1) \mathbf{S}^3 + \\ & (q^{10} + q^9 + 2q^8 + 2q^7 + 2q^6 + 2q^5 + 2q^4 + q^3 + q^2 + q) \mathbf{S}^{21} \end{aligned}$$

Le coefficient  $q^{12} + 3q^{11} + 4q^{10} + 5q^9 + 4q^8 + 4q^7 + 3q^6 + 2q^5 + q^4 + q^3$  de  $\mathbf{S}^{111}$  est :

$$\begin{aligned} \alpha_q(111) = & \mathbf{M}_1([3]_q) q^3 \mathbf{M}_1([2]_q) q^5 \mathbf{M}_1([2]_q) + \mathbf{M}_1([3]_q) q^6 \mathbf{M}_{11}([2]_q) + \\ & \mathbf{M}_{11}([3]_q) q^3 \mathbf{M}_1([4]_q) + \mathbf{M}_{111}([3]_q) \\ = & (q^{12} + 3q^{11} + 4q^{10} + 3q^9 + q^8) + (q^9 + q^8 + q^7) + \\ & (q^9 + 2q^8 + 3q^7 + 3q^6 + 2q^5 + q^4) + q^3 \end{aligned}$$

où le terme  $q^{12} + 3q^{11} + 4q^{10} + 3q^9 + q^8$  provient de la composition  $\tau = 111$ , le terme  $q^9 + q^8 + q^7$  provient de la composition  $12$ , le terme  $q^9 + 2q^8 + 3q^7 + 3q^6 + 2q^5 + q^4$  provient de la composition  $21$  et enfin  $q^3$  de la composition  $3$ .

En spécialisant  $q = 1$ , on retrouve

$$\mathbf{ch}(\mathcal{P}_\chi; n) = \sum_{\pi \models n} \alpha(\pi) \mathbf{S}^\pi \quad \text{avec} \quad \alpha(\pi) = \sum_{\tau \models \ell(\pi)} \prod_{i=1}^{\ell(\tau)} \binom{\Psi(\chi; \pi, \tau, i)}{\tau_i}. \quad [\text{PV15, Théorème 3.4}]$$

On sait interpréter chaque coefficient  $\alpha(\pi)$  comme l'ensemble des chemins sous la courbe  $\chi$  (§ 2.5). Cette première manière de calculer  $\alpha(\pi)$  est foncièrement inefficace (la complexité en temps est  $\Theta(2^{\ell(\pi)})$ ); en utilisant la technique d'inclusion-exclusion sur les manières de produire les chemins sous  $\chi$ , on obtient une seconde formule plus efficace :

**Proposition 4.2.9 :** Soit  $\pi = \pi_1 \pi_2 \cdots \pi_m$  une composition de  $n$  (avec  $n > 0$ ).

$$\alpha(\pi) = \sum_{k=1}^m (-1)^{k-1} \alpha(\pi_1 \pi_2 \cdots \pi_{m-k}) \binom{\chi(\pi(m-k) + 1) - m + k}{k},$$

et  $\alpha(\pi) = 1$  pour  $\pi$  l'unique composition de 0.

PREUVE : La formule est équivalente à

$$\sum_{k=0}^m (-1)^{k-1} \alpha(\pi_1 \pi_2 \cdots \pi_{m-k}) \binom{\chi(\pi(m-k) + 1) - m + k}{k} = 0.$$

Elle énumère des couples de chemins (comptés par  $\alpha(\pi)$ ) et d'ensemble de points (comptés par le binomial). Pour prouver la formule, on va construire une involution entre des couples signés différemment.

On suppose  $k$  fixé et on pose  $\gamma := \pi_1 \pi_2 \cdots \pi_{m-k}$ .

Le terme  $\alpha(\gamma)$  compte le nombre de chemins d'évaluation  $\gamma$  sous la courbe discrète  $\chi$ . On note un tel chemin par la séquence des ordonnées (des marches de longueur non nulles)  $C_{m-k} = (c_1, c_2, \dots, c_{m-k})$ .

Le binomial de la formule,  $\binom{\chi(\pi(m-k)+1)-m+k}{k}$ , dénombre les choix de  $k$  points entre 1 et  $\chi(\pi(m-k)+1)$  avec  $m-k$  points interdits. On note  $P_k$  un tel ensemble de points. L'ensemble des  $m-k$  points interdits correspond aux ordonnées du chemin  $C_{m-k}$ . À partir de chaque couple  $(C_{m-k}, P_k)$ , on peut donc reconstruire un chemin d'évaluation  $\pi$  sous la courbe discrète  $\chi$ .

On définit l'endofonction  $\zeta$  sur l'ensemble des couples  $(C_{m-k}, P_k)$  (avec  $0 \leq k \leq m$ ) :

- Si  $\max(P_k) =: p > c_{m-k}$  :

$$\zeta : (C_{m-k}, P_k) \mapsto (C_{m-k+1}, P_{k-1}), \quad (4.7)$$

où  $C_{m-k+1} := (c_1, c_2, \dots, c_{m-k}, p)$  et  $P_{k-1} := P_k \setminus \{p\}$ .

Par définition de  $C_{m-k}$  et de  $P_k$ , cette nouvelle séquence de  $m-k+1$  points,  $C_{m-k+1}$ , définit bien un chemin d'évaluation  $\pi_1 \cdots \pi_{m-k} \pi_{m-k+1}$  sous la courbe  $\chi$ . Le chemin  $C_{m-k+1}$  est donc bien un des  $\alpha(\pi_1 \pi_2 \cdots \pi_{m-k} \pi_{m-k+1})$  chemins d'évaluation  $\pi_1 \pi_2 \cdots \pi_{m-k+1}$ .

L'ensemble des points restants de  $P_{k-1}$  sont strictement plus petits que  $p$ . Comme  $\chi$  est croissante, on a alors

$$p \leq \chi(\pi(m-k)+1) \leq \chi(\pi(m-k+1)+1).$$

L'ensemble de points  $P_{k-1}$  est bien aussi un des ensembles de  $k-1$  points entre 1 et  $\chi(\pi(m-k+1)+1)$  distincts de  $p$  et de chaque  $c_i$ .

– sinon

$$\zeta : (C_{m-k}, P_k) \mapsto (C_{m-k-1}, P_{k+1}), \quad (4.8)$$

où  $C_{m-k-1} := (c_1, c_2, \dots, c_{m-k-1})$  et  $P_{k+1} := P_k \cup \{c_{m-k}\}$ .

Comme précédemment, on déduit facilement que  $C_{m-k-1}$  est un des chemins d'évaluation  $\pi_1 \pi_2 \cdots \pi_{m-k-1}$  sous la courbe  $\chi$  et que  $P_{k+1}$  est un des ensembles de  $k+1$  points entre 1 et  $\chi(\pi(m-k)+1)$  tous distincts des  $c_i$  restants de  $C_{m-k-1}$ .

On déduit de (4.7) et (4.8) que  $\zeta$  est une involution.

Dans la formule, la signature  $(-1)^{k-1}$  définit le signe de chaque couple  $(C_{m-k}, P_k)$  donc que chaque couple s'annule avec son image par  $\zeta$ . ■

```
sage: pi = (4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4)
sage: chi = lambda m: m^2 + m - 1
sage: # version somme de compositions (Théorème)
sage: %time alpha1(chi, pi)
CPU times: user 23.1 s, sys: 303 ms, total: 23.4 s
Wall time: 23.5 s
96325744652515379569481071134608164932
sage: # version Proposition
```

```
sage: %time alpha2(chi, pi)
CPU times: user 13.3 ms, sys: 1.36 ms, total: 14.6 ms
Wall time: 13.8 ms
96325744652515379569481071134608164932
```

### 4.3. Caractères dans d'autres bases

Après avoir exprimé et interprété le caractère des fonctions de parking généralisées dans les fonctions symétriques non-commutatives complètes, dans cette sous-section, on étudie le caractère des  $\chi$ -fonctions de parking dans d'autres bases que la base des fonctions complètes. Dans un premier paragraphe, on s'intéresse au développement du caractère dans la base des rubans :  $(\mathbf{R}_\pi)$  ; dans un second, on s'intéresse à la base des fonctions élémentaires  $(\Lambda^\pi)$ . On décrit l'interprétation combinatoire de chacun de ces développements.

#### 4.3.1. Fonctions rubans

À partir du changement de base sur les générateurs :  $\mathbf{S}_n = \mathbf{R}_n$ , on déduit immédiatement du Lemme 4.2.4 :

**Corollaire 4.3.1 :**

$$\mathbf{ch}_q(\mathcal{P}_\chi; n) = \sum_{\pi \models n} \prod_{i=1}^{\ell(\pi)} q^{\chi(1+\pi(i-1))\pi_i} \cdot \mathbf{R}_{\pi_i} \left( [\Upsilon(\chi; \pi, i)]_q \cdot \mathbb{A} \right).$$

**Exemple 4.3.2 :**

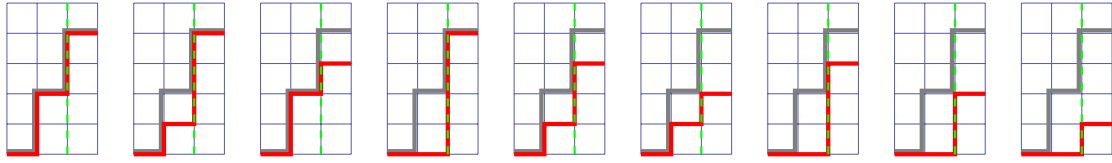
$$\begin{aligned} \mathbf{ch}_q(\mathcal{P}(2m-1); 3) = & (q^6 + 2q^5 + q^4 + q^3) \mathbf{R}_{111} + (q^6 + 2q^5 + 2q^4 + q^3 + q^2) \mathbf{R}_{12} + \\ & (q^6 + 2q^5 + 2q^4 + 2q^3 + q^2 + q) \mathbf{R}_{21} + \\ & (q^6 + 2q^5 + 3q^4 + 2q^3 + 2q^2 + q + 1) \mathbf{R}_3 \end{aligned}$$

**Remarque 4.3.3 :** Le coefficient des termes de queue  $(\mathbf{R}_n)$  encode l'ensemble des chemins escaliers sous  $\chi$ .

Soit  $\pi$  une composition de  $n$  de  $k$  parts. L'interprétation du coefficient  $\alpha_\pi$  de  $\mathbf{R}_\pi$  décrit l'ensemble des  $\chi$ -fonctions de parking dont les chemins ont certaines contremarches contraintes par  $\pi$ . La composition force l'existence d'une contre-marche à toutes les ordonnées qui coïncident avec une somme partielle  $\pi(j) < n$ . Autrement dit, chaque élément de l'ensemble des descentes de  $\pi$  est l'abscisse d'au moins une contre-marche.

Le coefficient  $\alpha_\pi$  du ruban  $\mathbf{R}_\pi$  encode l'ensemble des chemins ( $\chi$ -fonction de parking à isomorphisme près) ayant un pas vertical d'ordonnée  $\pi(1)$ , un pas vertical d'ordonnée  $\pi(2)$ , etc., jusqu'à  $\pi(k-1)$ .

**Exemple 4.3.4 :** Les coefficients  $q^6 + 2q^5 + 2q^4 + 2q^3 + q^2 + q$  de  $\mathbf{R}_{21}$  (Exemple 4.3.2) sont les 9 chemins (de la plus grande aire à la plus petite suivant l'ordre des coefficients) ayant un pas vertical d'ordonnée 2 (en vert hachuré) :



Du changement de base général :  $\mathbf{S}_\pi = \sum_{\pi \prec \tau} \mathbf{R}_\tau$  et du Théorème 4.2.7, on déduit :

**Corollaire 4.3.5 :**

$$\mathbf{ch}_q(\mathcal{P}_\chi; n) = \sum_{\pi \models n} \beta_q(\pi) \mathbf{R}_\pi \quad \text{avec} \quad \beta_q(\pi) = \sum_{\tau \prec \pi} \alpha_q(\tau).$$

En spécialisant cette  $q$ -formule avec  $q$  à 1, on retombe sur la formule sur les rubans présentée dans [PV15, § 3.3] :

$$\mathbf{ch}(\mathcal{P}_\chi; n) = \sum_{\pi \models n} \beta(\pi) \mathbf{R}_\pi \quad \text{avec} \quad \beta(\pi) = \sum_{\tau \prec \pi} \alpha(\tau).$$

**Exemple 4.3.6 :** On pose  $G(n) := \mathbf{ch}(\mathcal{P}(m^2 - m + 1); n)$ ,

$$G(0) = 1, \quad G(1) = \mathbf{R}_1, \quad G(2) = 2\mathbf{R}_{11} + 3\mathbf{R}_2,$$

$$G(3) = 9\mathbf{R}_{111} + 11\mathbf{R}_{12} + 15\mathbf{R}_{21} + 18\mathbf{R}_3,$$

$$G(4) = 70\mathbf{R}_{1111} + 79\mathbf{R}_{112} + 91\mathbf{R}_{121} + 102\mathbf{R}_{13} + 121\mathbf{R}_{211} + 136\mathbf{R}_{22} + 154\mathbf{R}_{31} + 172\mathbf{R}_4.$$

Les coefficients des termes de têtes ( $\mathbf{R}_{1^n}$ ) semblent être liés à [A177450] et les termes de queues ( $\mathbf{R}_n$ ) semblent être liés à [A177447].

### 4.3.2. Fonctions élémentaires

En reprenant l'expression du Lemme 4.2.4 en  $q = 1$ , à partir du caractère des séquences (2.5), on déduit l'expression suivante :

$$\mathbf{ch}(\mathcal{P}_\chi; n) = \sum_{\pi \models n} \prod_{i=1}^{\ell(\pi)} \mathbf{ch}(\mathbf{E}^{\Upsilon_\pi^{(i)}}; \pi_i),$$

(où on reprend la notation  $\Upsilon_\pi^{(i)}$  au lieu de  $\Upsilon(\chi; \pi, i)$ )

$$\mathbf{ch}(\mathcal{P}_\chi; n) = \sum_{\pi \models n} \prod_{i=1}^{\ell(\pi)} \left( \sum_{\tau \models \pi_i} (-1)^{\pi_i - \ell(\tau)} \binom{\Upsilon_\pi^{(i)} - 1 + \ell(\tau)}{\ell(\tau)} \Lambda^\tau \right),$$



$$= \sum_{\pi \models n} \sum_{\substack{\tau_1 \models \pi_1 \\ \tau_2 \models \pi_2 \\ \vdots}} (-1)^{n - \sum \ell(\tau_i)} \binom{\Upsilon_{\pi}^{(1)} - 1 + \ell(\tau_1)}{\ell(\tau_1)} \binom{\Upsilon_{\pi}^{(2)} - 1 + \ell(\tau_2)}{\ell(\tau_2)} \dots \Lambda^{\tau_1 \tau_2 \dots},$$

comme pour l'expression dans la base des fonctions complètes, cette double somme encode toutes les manières de factoriser la séquence des parts de toutes les compositions  $\pi \models n$ . On obtient alors

$$\begin{aligned} \mathbf{ch}(\mathcal{P}_{\chi}; n) &= \sum_{\pi \models n} \sum_{\substack{\tau \models \ell(\pi) \\ \tau = \tau_1 \dots \tau_k}} (-1)^{n - \ell(\pi)} \binom{\chi(1) - 1 + \tau_1}{\tau_1} \binom{(\chi(1 + \pi(\tau(1)))) - \chi(1) - 1 + \tau_2}{\tau_2} \\ &\quad \dots \binom{[\chi(1 + \pi(\tau(k-1)))] - \chi(1 + \pi(\tau(k-2)))] - 1 + \tau_k}{\tau_k} \Lambda^{\pi} \end{aligned}$$

On en déduit le théorème :

**Théorème 4.3.7 :**

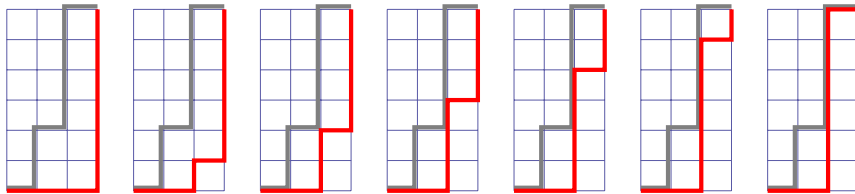
$$\mathbf{ch}(\mathcal{P}_{\chi}; n) = \sum_{\pi \models n} (-1)^{n - \ell(\pi)} \rho(\pi) \Lambda^{\pi} \quad \text{avec} \quad \rho(\pi) = \sum_{\tau \models \ell(\pi)} \prod_{i=1}^{\ell(\tau)} \binom{\Psi(\chi; \pi, \tau, i) - 1 + \tau_i}{\tau_i}.$$

On rappelle la définition de l'application  $\Psi$  est donnée Équation 4.6.

**Exemple 4.3.8 :** On pose  $G(n) := \mathbf{ch}(\mathcal{P}(m^2 - m + 1); n)$  :

$$\begin{aligned} G(0) &= 1, \quad G(1) = \Lambda^1, \quad G(2) = 3\Lambda^{11} - \Lambda^2, \\ G(3) &= 18\Lambda^{111} - 3\Lambda^{12} - 7\Lambda^{21} + \Lambda^3, \\ G(4) &= 172\Lambda^{1111} - 18\Lambda^{112} - 36\Lambda^{121} + 3\Lambda^{13} - 70\Lambda^{211} + 7\Lambda^{22} + 13\Lambda^{31} - \Lambda^4. \end{aligned}$$

Voici les 7 chemins associés au terme  $\Lambda^{21}$  (cf. l'interprétation § 2.7) :



## 4.4. Formules d'énumération

Ces différentes formules de caractères non-commutatifs se redescendent naturellement dans les fonctions symétriques. D'après le Théorème 1.5.11, on en déduit des formules d'énumération des  $\chi$ -fonctions de parking et de leurs types d'isomorphismes, les  $\chi$ -chemins.

### 4.4.1. Énumération de $\chi$ -fonctions de parking

À partir du caractère, on déduit les formules d'énumération des fonctions de parking :

$$p(\chi; n) = \sum_{k=1}^n \binom{n}{k} \chi(1)^k p(\psi_k; n-k), \quad [\text{Éq. 2.3 et 4.4}]$$

avec  $p(\chi; 0) = 1$  et où on rappelle  $\psi_k$  est le  $k$ -décalage de  $\chi$  (4.2)

$$p(\chi; n) = \sum_{\pi \models n} \prod_{i=1}^{\ell(\pi)} \Upsilon(\chi; \pi, i)^{\pi_i}, \quad [\text{Éq. 2.3 et Lemme 4.2.4}]$$

$$= \sum_{\substack{\pi \models n \\ \pi = \pi_1 \cdots \pi_k}} \alpha(\pi) \binom{n}{\pi_1, \dots, \pi_k}, \quad [\text{Éq. 2.3 et Théorème 4.2.7}]$$

$$= \sum_{\substack{\pi \models n \\ \pi = \pi_1 \cdots \pi_k}} (-1)^{n-\ell(\pi)} \rho(\pi) \binom{n}{\pi_1, \dots, \pi_k}. \quad [\text{Éq. 2.3 et Théorème 4.3.7}]$$

Si  $\chi(m) = b\psi(m)$  (avec  $b \geq 1$ ), pour tout  $m$ , alors on retrouve, par récurrence, l'égalité de la première formule :

$$p(\chi; n) = b^n p(\psi; n). \quad [\text{KY03, Corollaire 5.6}]$$

PREUVE: Par définition, on a  $p(\chi; 0) = b^0 p(\psi; 0)$ . D'où par récurrence,

$$\begin{aligned} p(\chi; n) &= \sum_{k=1}^n \binom{n}{k} b^k \psi(1)^k b^{n-k} p(\psi; n-k), \\ &= b^n \sum_{k=1}^n \binom{n}{k} \psi(1)^k p(\psi; n-k), \\ &= b^n p(\psi; n). \end{aligned} \quad \blacksquare$$

Cette égalité permet de déduire les suites et formules suivantes :

$\chi$	$p(\chi; n)$ pour $n = 0, 1, 2, 3, \dots$	Formule	OEIS
$m$	1, 1, 3, 16, 125, 1296, 16807, 262144, 4782969, 100000000, ...	$(n+1)^{n-1}$	<a href="#">A000272</a>
$2m$	1, 2, 12, 128, 2000, 41472, 1075648, 33554432, ...	$2^n (n+1)^{n-1}$	<a href="#">A097629</a>
$3m$	1, 3, 27, 432, 10125, 314928, 12252303, 573308928, ...	$3^n (n+1)^{n-1}$	<a href="#">A136719</a>
$\vdots$		$\vdots$	
$bm$		$b^n (n+1)^{n-1}$	

Cette égalité permet de déduire aussi automatiquement la relation suivante sur les séries génératrices :

$$\mathcal{P}_\chi(t) = \sum_{n \geq 0} p(\psi; n) b^n t^n = \mathcal{P}_\psi(bt). \quad (4.9)$$

Les  $\chi$ -fonctions de parking associées sont des cas particuliers bien étudiées dans [Sta99 ; Yan00 ; Yan01].

#### 4.4.2. Énumération de $\chi$ -chemins

On déduit, de la même manière, les formules d'énumération des types d'isomorphismes des fonctions de parking, c'est-à-dire des chemins sous la courbe  $\chi$  :

$$\begin{aligned} \tilde{p}(\chi; n) &= \sum_{k=1}^n \binom{k + \chi(1) - 1}{\chi(1) - 1} \tilde{p}(\psi_k; n - k), & [\text{Éq. 2.4 et 4.4}] \\ &= \sum_{\pi \models n} \prod_{i=1}^{\ell(\pi)} \binom{\pi_i + \Upsilon(\chi; \pi, i) - 1}{\Upsilon(\chi; \pi, i) - 1}, & [\text{Éq. 2.4 et Lemme 4.2.4}] \\ &= \sum_{\pi \models n} \alpha(\pi), & [\text{Éq. 2.4 et Théorème 4.2.7}] \\ &= \sum_{\pi \models n} (-1)^{n - \ell(\pi)} \rho(\pi). & [\text{Éq. 2.4 et Théorème 4.3.7}] \end{aligned}$$

Ces formules correspondent donc à l'énumération de l'ensemble des chemins sous la courbe discrète  $\chi$ . On peut retrouver ce nombre par inclusion-exclusion :

$$\tilde{p}(\chi; n) = \sum_{k=0}^{n-1} (-1)^{n-k+1} \tilde{p}(\chi; k) \binom{\chi(k+1)}{n-k},$$

avec  $\tilde{p}(\chi; 0) = 1$ .

PREUVE : La formule est équivalente à

$$\sum_{k=0}^n (-1)^{n-k+1} \tilde{p}(\chi; k) \binom{\chi(k+1)}{n-k} = 0. \quad (4.10)$$

Comme dans la preuve de la Proposition 4.2.9, on construit une involution sur des ensembles de couples  $(C, P)$  où  $C := (c_1, \dots, c_{\chi(k)})$  est un chemin de longueur  $k$  sous  $\chi$  (compté dans  $\tilde{p}(\chi; k)$ ) et  $P$  un ensemble de  $n - k$  points entre 1 et  $\chi(k+1)$  (compté par le binomial).

Soit  $C := (c_1, \dots, c_{\chi(k)})$  une décomposition de  $k$  en  $\chi(k)$  parts (qui encode un  $\chi$ -chemin) telle que

$$\sum_{i=1}^m c_i \geq m, \quad \text{pour tout } 1 \leq m \leq k.$$

Cette décomposition encode exactement (de manière unique) un chemin sous la courbe  $\chi$  de longueur  $k$ . Soit  $P$  un sous-ensemble de  $n - k$  entiers de  $[\chi(k + 1)]$ . (On dit que  $(C, P)$  est un couple pour  $k$  compter par  $\tilde{p}(\chi; k) \binom{\chi(k+1)}{n-k}$ .)

On pose l'application :

- Si  $k > 0$  et tous les entiers  $j \in P$  sont strictement plus petits que tous les entiers  $i$  pour lesquels  $c_i > 0$ , alors on pose  $m := \max\{i \mid c_i > 0\}$  et

$$(C, P) \mapsto ((c'_1, \dots, c'_{\chi(k-1)}), P \cup \{m\}) ,$$

avec  $c'_j = c_j$  si  $m \neq j$  et  $c'_m = c_m - 1$  sinon ;

- sinon

$$(C, P) \mapsto ((c'_1, \dots, c'_{\chi(k+1)}), P \setminus \{\max P\}) ,$$

avec

$$c'_j := \begin{cases} c_j + 1 & \text{si } j = \max P, \\ c_j & \text{si } j \leq \chi(k), \\ 0 & \text{sinon.} \end{cases}$$

L'application est une involution. Dans les deux cas, les couples obtenus sont des couples de chemin et ensemble comptés respectivement par

$$\tilde{p}(\chi; k - 1) \binom{\chi(k + 2)}{n - k + 1} \quad \text{ou} \quad \tilde{p}(\chi; k + 1) \binom{\chi(k)}{n - k - 1} .$$

Les signes associés dans la formule à ces couples sont inverses et donc chaque couple  $(C, P)$  s'annule avec son image par l'involution. On en déduit ainsi la formule (4.10).■

À partir de ces formules, voici quelques suites associées  $\chi$ -chemins :

$\chi$	$\tilde{p}(\chi; n)$ pour $n = 0, 1, 2, 3, \dots$	Formule	OEIS
$m$	1, 1, 2, 5, 14, 42, 132, 429, 1430, 4862, ...	$\binom{2n}{n}/n+1$	<a href="#">A000108</a>
$2m$	1, 2, 7, 30, 143, 728, 3876, 21318, 120175, ...	$\binom{3n+1}{n}/n+1$	<a href="#">A006013</a>
$3m$	1, 3, 15, 91, 612, 4389, 32890, 254475, 2017356, ...	$\binom{4n+2}{n}/n+1$	<a href="#">A006632</a>
$4m$	1, 4, 26, 204, 1771, 16380, 158224, 1577532, 16112057, ...	$\binom{5n+3}{n}/n+1$	<a href="#">A118971</a>
$2m + 1$	1, 3, 12, 55, 273, 1428, 7752, 43263, 246675, 1430715, ...	$\binom{3n+3}{n+1}/2n+3$	<a href="#">A001764</a>
$2m + 2$	1, 4, 18, 88, 455, 2448, 13566, 76912, 444015, 2601300, ...	$2 \binom{3n-3}{n-2}/n$	<a href="#">A006629</a>

## 5. Tables de parking et arborescences

Dans le chapitre précédent, on a redéfini les fonctions de parking (généralisées) à partir d'une grammaire (4.3) sur les séquences d'ensembles. Cette grammaire se définit explicitement à partir de l'espèce des ensembles  $\mathbf{E}$ . Dans ce chapitre, on remplace cette dernière par une espèce quelconque  $\mathbf{F}$ .

En terme informatique, les  $\chi$ -fonctions de parking définissent des fonctions de hachages d'un ensemble  $U$  de  $n$  clefs. À chaque clef  $u \in U$ , la  $\chi$ -fonction de parking associe une position favorite dans une table (de hachage) abstraite de  $\chi(n+1)$  indexes. En remplaçant  $\mathbf{E}$  par une espèce  $\mathbf{F}$ , on fixe une structure concrète de table de hachage (ayant pour fonction de hachage les  $\chi$ -fonctions de parking). Par exemple en utilisant l'espèce des ordres linéaires  $\mathcal{L}$ , on va réaliser toutes les instances de tables de hachage en listes.

Les idées du début de ce chapitre sont très simples. On reprend les formules du chapitre précédent en les adaptant pour obtenir des formules d'énumération des différentes instances de tables de hachage, comme celles des tables de hachage en liste.

Dans une première section, on définit cette espèce de  $\chi$ -tables de parking et on donne de nouvelles formules d'énumération. Dans les sections suivantes, on étudie les cas particuliers où  $\chi$  est associée à une fonction affine. Nous décrivons une bijection générique avec des structures arborescentes qui généralisent les travaux de Catherine H. YAN [Yan00 ; Yan01]. Dans une seconde section, on rappelle quelques bijections usuelles entre les fonctions de parking et les arbres enracinés (ou forêts d'arbres enracinés) parmi les nombreuses existantes dans la littérature [Sch68 ; Rio69 ; FR74 ; Kre80 ; Yan00 ; Sta04 ; Shi08 ; GL11]. Enfin dans une troisième section, on décrit une bijection générique entre les  $\chi$ -tables de parking et des structures arborescentes dans le cas où  $\chi$  est affine ; cette dernière généralise naturellement la bijection de C. H. YAN.

### 5.1. Tables de parking (généralisées)

Dans le chapitre précédent, on a décrit une grammaire (4.3) qui construit toute une famille de fonctions de hachages : les  $\chi$ -fonctions de parking. Dans cette section, on cherche à construire l'ensemble des structures (ou instances) de différentes implémentations de tables de hachage en remplaçant l'espèce  $\mathbf{E}$  par une espèce  $\mathbf{F}$  encodant l'implémentation de la table.

### 5.1.1. Définition

Soit  $\mathbf{F}$  une espèce et soit  $\chi : \mathbb{N}_{>0} \rightarrow \mathbb{N}$  une suite croissante.

**Remarque 5.1.1 :** On suppose les  $\chi$ -fonctions de parking sur  $U$  (de cardinalité  $n$ ),  $\mathcal{P}_\chi[U]$ , définies comme les séquences de  $\chi(n+1)$  ensembles, comme décrites par la grammaire (4.3). On rappelle que, dans le cas des fonctions de parking, le fait de considérer des séquences de longueur  $\chi(n)$  ou  $\chi(n+1)$  n'est pas important. En effet  $\mathbf{E}[\emptyset]$  est un singleton contenant uniquement l'ensemble vide. Dans les cas suivants de ce chapitre, l'ensemble des structures sur l'ensemble vide n'est pas nécessairement un singleton et cela va jouer un rôle dans le cadre des formules d'énumération, en particulier pour les bijections avec les structures arborescentes (cf. § 5.3).

On définit l'espèce des  $\chi$ -tables de parking en  $\mathbf{F}$  comme l'espèce  $\triangleright_\chi^{\mathbf{F}}$  des séquences  $(f_i)$  de  $\chi(n+1)$  structures de l'espèce  $\mathbf{F}$ , sur un ensemble  $U$  de  $n$  éléments :

$$\triangleright_\chi^{\mathbf{F}}[U] := \{ (f_i)_{i \in [\chi(n+1)]} \text{ telles que pour tout } i, f_i \in \mathbf{F}[U_i] \text{ et } (U_i) \in \mathcal{P}_\chi[U] \},$$

et, où pour toute bijection  $\sigma : U \rightarrow V$  :

$$\triangleright_\chi^{\mathbf{F}}[\sigma] : (f_i) \longmapsto (\mathbf{F}[\sigma|_{U_i}](f_i)).$$

**Remarque 5.1.2 :** Les  $\chi$ -fonctions de parking sont des  $\chi$ -tables de parking en  $\mathbf{E}$ , c'est-à-dire  $\mathcal{P}(\chi) = \triangleright_\chi^{\mathbf{E}}$ .

**Exemple 5.1.3 :** Les trente  $m$ -tables de parking en  $\mathcal{L}$  sur  $[3]$  sont :

$$\begin{aligned} & (123 \mid \cdot \mid \cdot \mid \cdot), (132 \mid \cdot \mid \cdot \mid \cdot), (312 \mid \cdot \mid \cdot \mid \cdot), (213 \mid \cdot \mid \cdot \mid \cdot), (231 \mid \cdot \mid \cdot \mid \cdot), (321 \mid \cdot \mid \cdot \mid \cdot), \\ & (12 \mid 3 \mid \cdot \mid \cdot), (12 \mid \cdot \mid 3 \mid \cdot), (21 \mid 3 \mid \cdot \mid \cdot), (21 \mid \cdot \mid 3 \mid \cdot), (13 \mid 2 \mid \cdot \mid \cdot), (13 \mid \cdot \mid 2 \mid \cdot), \\ & (31 \mid 2 \mid \cdot \mid \cdot), (31 \mid \cdot \mid 2 \mid \cdot), (23 \mid 1 \mid \cdot \mid \cdot), (23 \mid \cdot \mid 1 \mid \cdot), (32 \mid 1 \mid \cdot \mid \cdot), (32 \mid \cdot \mid 1 \mid \cdot), \\ & (1 \mid 23 \mid \cdot \mid \cdot), (1 \mid 32 \mid \cdot \mid \cdot), (3 \mid 21 \mid \cdot \mid \cdot), (3 \mid 12 \mid \cdot \mid \cdot), (2 \mid 13 \mid \cdot \mid \cdot), (2 \mid 31 \mid \cdot \mid \cdot), \\ & (1 \mid 2 \mid 3 \mid \cdot), (1 \mid 3 \mid 2 \mid \cdot), (3 \mid 1 \mid 2 \mid \cdot), (2 \mid 1 \mid 3 \mid \cdot), (2 \mid 3 \mid 1 \mid \cdot), (3 \mid 2 \mid 1 \mid \cdot). \end{aligned}$$

À partir du Lemme 4.1.6, cette espèce  $\triangleright_\chi^{\mathbf{F}}$  peut se définir de manière équivalente à par la grammaire :

$$\triangleright_\chi^{\mathbf{F}} = (\mathbf{F}^{\chi(1)})_0 + \sum_{n \geq 1} (\mathbf{F}^{\chi(1)})_n \cdot \triangleright_{\psi_n}^{\mathbf{F}}. \quad (5.1)$$

### 5.1.2. Série indicatrice de cycles

Cette grammaire permet d'obtenir automatiquement que la série indicatrice de cycle de  $\triangleright_\chi^{\mathbf{F}}$  satisfait l'équation :

$$Z_{\triangleright_\chi^{\mathbf{F}}} = \text{ch}(\mathbf{F}^{\chi(1)}; 0) + \sum_{n \geq 1} \text{ch}(\mathbf{F}^{\chi(1)}; n) Z_{\triangleright_{\psi_n}^{\mathbf{F}}}. \quad (5.2)$$

On en déduit :

$$\text{ch}(\triangleright_{\chi}^{\mathbf{F}}; n) = \sum_{k=1}^n \text{ch}(\mathbf{F}^{\chi(1)}; k) \text{ch}(\triangleright_{\psi_k}^{\mathbf{F}}; n - k) ,$$

pour  $n \geq 1$  et  $\text{ch}(\triangleright_{\chi}^{\mathbf{F}}; 0) = \text{ch}(\mathbf{F}; 0)^{\chi(1)}$ .

En développant la récurrence, on déduit une formule analogue à celle du Lemme 4.2.4 (en  $q = 1$ ) :

$$\text{ch}(\triangleright_{\chi}^{\mathbf{F}}; n) = \left( \sum_{\pi \models n} \prod_{i=1}^{\ell(\pi)} \text{ch}(\mathbf{F}^{\Upsilon(\chi; \pi, i)}; \pi_i) \right) \text{ch}(\mathbf{F}^{\Upsilon(\chi; \pi, \ell(\pi)+1)}; 0) . \quad (5.3)$$

Le terme supplémentaire permet de prendre en compte le cas où  $\text{ch}(\mathbf{F}; 0) \neq 1$  et ainsi la queue de la séquence de  $\mathbf{F}$ -structures sur l'ensemble vide.

**Remarque 5.1.4 :**

$$\text{ch}(\mathbf{F}^{\Upsilon(\chi; \pi, \ell(\pi)+1)}; 0) = \text{ch}(\mathbf{F}; 0)^{\Upsilon(\chi; \pi, \ell(\pi)+1)} .$$

**Exemple 5.1.5 :** En reprenant l'exemple précédent avec l'espèce des ordres linéaires, on a alors :

$$\begin{aligned} \text{ch}(\triangleright_{\chi}^{\mathcal{L}}; n) &= \sum_{k=1}^n \text{ch}(\mathcal{L}^{\chi(1)}; n) \text{ch}(\triangleright_{\psi_n}^{\mathcal{L}}; n - k) & [\text{Éq. 5.2}] \\ &= \sum_{k=1}^n [n] \left( (1 - h_1)^{-\chi(1)} \right) \text{ch}(\triangleright_{\psi_n}^{\mathcal{L}}; n - k) & [\text{Éq. 1.18}] \\ &= \sum_{k=1}^n \binom{k + \chi(1) - 1}{k} h_1^k \text{ch}(\triangleright_{\psi_n}^{\mathcal{L}}; n - k) & [\text{Éq. 2.4}] \\ &= h_1^n \sum_{\pi \models n} \prod_{i=1}^{\ell(\pi)} \binom{\pi_i + \Upsilon(\chi; \pi, i) - 1}{\pi_i} , & [\text{Éq. 5.3}] \end{aligned}$$

avec  $\text{ch}(\mathcal{L}^{\chi(n+1)-\chi(n)}; 0) = 1$ . La somme dans l'équation précédente correspond à la spécialisation  $\mathbf{S}^{\pi} \rightarrow 1$  du Lemme 4.2.4 (pour  $q = 1$ ). Dans le cas où  $\chi$  est l'identité, on sait que cette somme sur les compositions de  $n$  est égale au  $n^{\text{ième}}$  nombre de Catalan. On a alors

$$\text{ch}(\triangleright_{\text{Id}}^{\mathcal{L}}; n) = h_1^n C_n .$$

Par spécialisation (Théorème 1.5.11), on déduit alors que les  $\text{Id}$ -tables de parking en  $\mathcal{L}$  et leurs types d'isomorphisme sont respectivement en bijection avec les arbres binaires étiquetés et non-étiquetés.

**Exemple 5.1.6 :** On considère l'espèce des ensembles bicolores  $2\mathbf{E}$  (ou  $\mathbf{E} + \mathbf{E}$ ). La série indicatrice de cette espèce est

$$Z_{2\mathbf{E}} = \sum_{n \geq 0} 2h_n = 2\sigma_1(\mathbb{X}).$$

Dans le cas où  $\chi$  est l'identité, en appliquant la technique de H. O. POLLAK (§ 3.4), on déduit immédiatement

$$\text{ch}(\triangleright_{\text{Id}}^{2\mathbf{E}}; n) = 2^{n+1} \text{ch}(\mathcal{P}; n) = 2^{n+1} \frac{\text{ch}(\mathbf{E}^{n+1}; n)}{n+1}.$$

**Remarque :** Si on considère uniquement les séquences de taille  $n$  alors il semble exister une bijection entre les structures de partitions d'ensembles décrites par [A216794] où les coefficients de la série associée sont simplement :  $2^n(n+1)^{n-1}$  (au lieu de  $2^{n+1}(n+1)^{n-1}$ ).

Évidemment quelque soit  $k \geq 1$ , on a l'identité :

$$\text{ch}(\triangleright_{\chi}^{k\mathbf{E}}; n) = k^{\chi(n+1)} \text{ch}(\mathcal{P}(\chi); n).$$

De manière encore plus générale que dans l'exemple précédent, on a :

**Proposition 5.1.7 :** Soit  $k \geq 1$ .

$$\text{ch}(\triangleright_{\chi}^{k\mathbf{F}}; n) = k^{\chi(n+1)} \text{ch}(\triangleright_{\chi}^{\mathbf{F}}; n).$$

PREUVE: Par définition quelque soit l'espèce  $\mathbf{G}$ ,

$$\begin{aligned} Z_{(k\mathbf{G})^n} &= (Z_{k\mathbf{G}})^n, & [Z_{\mathbf{F} \cdot \mathbf{G}} &= Z_{\mathbf{F}} Z_{\mathbf{G}}] \\ &= (k Z_{\mathbf{G}})^n, & [Z_{\mathbf{F} + \mathbf{G}} &= Z_{\mathbf{F}} + Z_{\mathbf{G}}] \\ &= k^n Z_{\mathbf{G}}^n = k^n Z_{\mathbf{G}^n}. \end{aligned}$$

En particuliers,

$$\begin{aligned} \text{ch}(\triangleright_{\chi}^{k\mathbf{F}}; 0) &= \text{ch}((k\mathbf{F})^{\chi(1)}; 0) & [\text{Éq. 5.1}] \\ &= k^{\chi(1)} \text{ch}(\mathbf{F}^{\chi(1)}; 0) = k^{\chi(1)} \text{ch}(\triangleright_{\chi}^{\mathbf{F}}; 0), \end{aligned}$$

d'où par récurrence, on déduit alors

$$\begin{aligned} \text{ch}(\triangleright_{\chi}^{k\mathbf{F}}; n) &= \sum_{j=1}^n \text{ch}((k\mathbf{F})^{\chi(1)}; j) \text{ch}(\triangleright_{\psi_j}^{k\mathbf{F}}; n-j), \\ &= \sum_{j=1}^n k^{\chi(1)} \text{ch}((\mathbf{F})^{\chi(1)}; j) k^{\psi_j(n-j+1)} \text{ch}(\triangleright_{\psi_j}^{\mathbf{F}}; n-j), & [\text{hypothèse}] \\ &= k^{\chi(n+1)} \text{ch}(\triangleright_{\chi}^{\mathbf{F}}; n). & [\text{Éq. 4.2}] \end{aligned}$$

■



### 5.1.3. Formules d'énumération

En reprenant les caractères précédents, on déduit que les formules d'énumérations des structures et des types d'isomorphismes satisfont :

$$\begin{aligned} q(\triangleright_{\chi}^F; n) &= \sum_{j=1}^n \binom{n}{j} [t^j] (F(t)^{\chi(1)}) q(\triangleright_{\psi_j}^F; n-j), \\ \tilde{q}(\triangleright_{\chi}^F; n) &= \sum_{j=1}^n [t^j] (\tilde{F}(t)^{\chi(1)}) \tilde{q}(\triangleright_{\psi_j}^F; n-j), \end{aligned} \quad [\text{Éq. 5.2}]$$

respectivement avec  $q(\triangleright_{\chi}^F; 0) = [t^0] (F(t)^{\chi(1)})$  et  $\tilde{q}(\triangleright_{\chi}^F; 0) = [t^0] (\tilde{F}(t)^{\chi(1)})$ . Ces récurrences se déroulent respectivement comme les sommes sur les compositions de  $n$  suivantes :

$$\begin{aligned} q(\triangleright_{\chi}^F; n) &= \left( \sum_{\substack{\pi \models n \\ \pi = \pi_1 \cdots \pi_k}} \binom{n}{\pi_1, \dots, \pi_k} \prod_{i=1}^k [t^{\pi_i}] (F(t)^{\Upsilon(\chi; \pi, i)}) \right) ([t^0] F(t))^{\Upsilon(\chi; \pi, k+1)}, \\ \tilde{q}(\triangleright_{\chi}^F; n) &= \left( \sum_{\substack{\pi \models n \\ \pi = \pi_1 \cdots \pi_k}} \prod_{i=1}^k [t^{\pi_i}] (\tilde{F}(t)^{\Upsilon(\chi; \pi, i)}) \right) ([t^0] \tilde{F}(t))^{\Upsilon(\chi; \pi, k+1)}. \end{aligned} \quad [\text{Éq. 5.3}]$$

À partir du Corollaire 4.2.5 (ou [KY03, Corollaire 5.6]) et la Proposition 5.1.7, on déduit l'égalité suivante :

$$q(\triangleright_{\chi}^{kE}; n) = k^{\chi(n+1)} p(\chi; n) = k^{\chi(n+1)-n} p(k\chi; n).$$

### 5.1.4. Cas particulier : identité

Dans le cas particulier où  $\chi$  est l'identité, la preuve de Henry O. POLLAK (présentée dans [Rio69 ; FR74] et abordée ici dans la section § 3.4) s'applique encore aux tables de parking quelque soit l'espèce  $F$  considérée. En effet, cette preuve ne dépend que des  $n+1$  emplacements et des  $n$  étiquettes. Autrement dit, on a la formule d'énumération :

$$q(\triangleright_{\text{Id}}^F; n) = \frac{[t^n] (F(t)^{n+1})}{n+1}.$$

À partir de cette égalité, on peut donc calculer efficacement ce nombre en utilisant n'importe quel algorithme d'exponentiation rapide pour calculer  $F(t)^{n+1}$  (de tels algorithmes sont abordés dans [Knu81]).

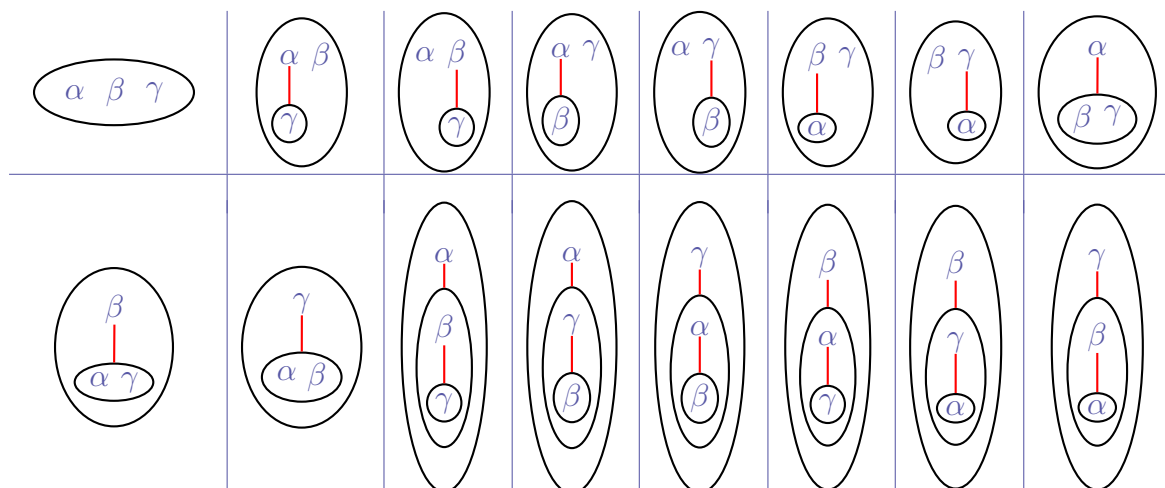
## 5.2. Forêts d'arbres enracinés

Les fonctions de parking sont bien connues pour être en bijection avec les *forêts d'arbres enracinés*. Il existe de nombreuses bijections dans la littérature, on peut notamment citer [Sch68 ; Rio69 ; FR74 ; Kre80 ; Sta04 ; Shi08 ; GL11].

Ces forêts sont les structures de l'espèce solution de l'équation fonctionnelle (1.20) :

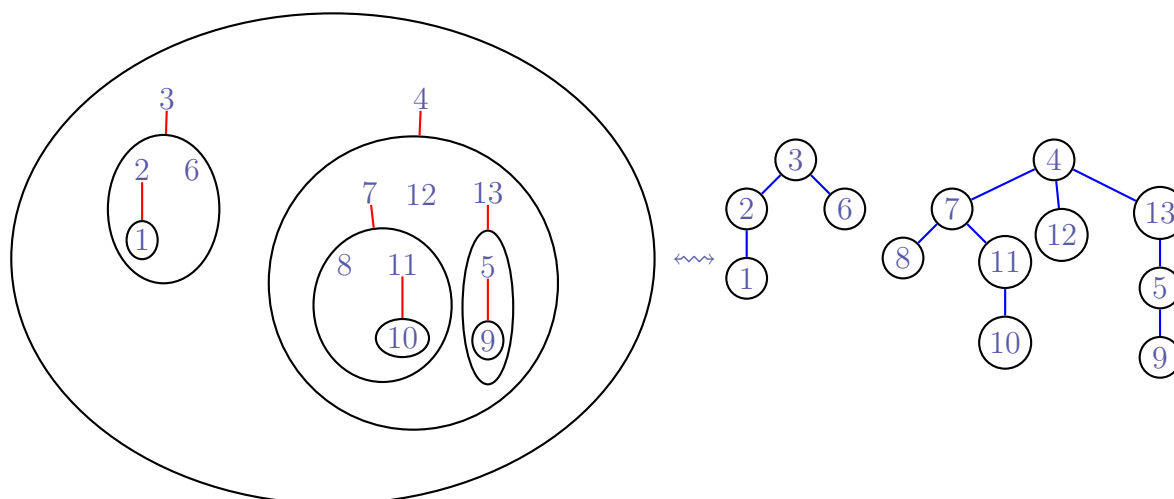
$$\mathcal{F} = \mathbb{E}(X \cdot \mathcal{F}).$$

**Exemple 5.2.1 :** Voici les seize forêts d'arbres sur l'ensemble  $\{\alpha, \beta, \gamma\}$  :



Chaque forêt est représentée/(définie récursivement) comme un ensemble racine (l'ellipse externe) composé d'étiquettes (non incluses dans les ellipses internes) rattachées (en rouge) à une forêt (les ellipses les plus externes à l'intérieur de l'ellipse externe). La forêt vide est représentée par l'absence d'arête (et d'ensemble).

Pour se ramener à une structure « plus arborescente », il suffit de considérer l'ensemble des étiquettes appartenant uniquement à l'ellipse la plus externe comme racines, et, à chaque racine  $\alpha$  on associe, comme fils, l'ensemble des racines de la forêt définie récursivement à partir de l'ensemble rattaché à  $\alpha$  par l'arête rouge :



### 5.2.1. Énumération des forêts d'arbres enracinés

Le nombre de forêts d'arbres enracinés se déduit aisément de la formule de Arthur CALEY qui dénombre l'ensemble des arbres (non-enracinés) à  $n$  nœuds :

$$T_n = n^{n-2}. \quad [\text{Formule de Caley, 1889}]$$

Pour se ramener à l'énumération des forêts d'arbres enracinés, il suffit de fixer un nœud  $q$  particulier (par exemple celui ayant l'étiquette maximale dans le cas où l'ensemble d'étiquette est  $[n]$ ), et, de considérer l'unique forêt d'arbres enracinés définie à partir des nœuds fils de  $q$ . On a donc :

$$[t^n]\mathcal{F}(t) = T_{n+1} = (n+1)^{n-1}.$$

En terme d'espèces de structures [BLL98], ce résultat découle de l'égalité suivante :

$$\mathcal{F} = \mathcal{a}',$$

où  $\mathcal{a}$  désigne l'espèce des *arbres* (ou *graphes simples sans cycle*) et où  $\mathcal{F}'$  correspond à la dérivée de l'espèce  $\mathcal{F}$  [BLL98, § 2.2.2].

La formule de Caley s'obtient de différentes manières, en voici quelques-unes très élégantes :

#### Théorème des matrices-arbres

Provenant de la théorie des graphes, la démonstration la plus classique est certainement celle issue du théorème de Gustav R. KRICHHOFF en 1847. Cette première démonstration est un corollaire du *théorème des matrices-arbres* qui est un théorème très puissant permettant de compter l'ensemble des arbres couvrants d'un graphe quelconque. (Le théorème est donné ici sans détail et sans preuve afin d'aller à l'essentiel pour retrouver la formule de Caley.)

Une matrice-arbres est une matrice Laplacienne définie comme la différence entre la matrice des degrés et la matrice d'adjacence d'un graphe quelconque. Soit  $G = (V, E)$  un graphe. La matrice-arbres  $L$  de  $G$  est la matrice définie par :

$$L_{i,j} := \begin{cases} \deg(v_i) & \text{si } i = j, \\ -1 & \text{si } i \neq j \text{ et } (v_i, v_j) \in E, \\ 0 & \text{sinon.} \end{cases}$$

**Théorème 5.2.2 (Kirchhoff, 1847) :** Le nombre  $t(G)$  d'arbres couvrants du graphe  $G$  est égal, au signe près, à la valeur de n'importe quel cofacteur de  $L$ .

Ce théorème permet d'énumérer l'ensemble des sous-graphes de  $G$  identifiable à des arbres. La formule de Caley s'obtient donc comme un cas particulier où le graphe considéré est un graphe complet  $K_n$  (contenant ainsi tous les arbres possibles). La matrice

Laplacienne  $n \times n$  est alors :

$$L = \begin{pmatrix} n-1 & -1 & -1 & \cdots & -1 \\ -1 & n-1 & -1 & \cdots & -1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ -1 & -1 & \cdots & n-1 & -1 \\ -1 & -1 & -1 & \cdots & n-1 \end{pmatrix}.$$

En supprimant par exemple la première ligne et la première colonne, on obtient le cofacteur  $(n-1) \times (n-1)$  de même forme ayant pour déterminant :

$$\det(L^*) = \begin{vmatrix} n-1 & -1 & -1 & \cdots & -1 \\ -1 & n-1 & -1 & \cdots & -1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ -1 & -1 & \cdots & n-1 & -1 \\ -1 & -1 & -1 & \cdots & n-1 \end{vmatrix} = \begin{vmatrix} 1 & 0 & \cdots & 0 & 0 \\ -1 & n & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ -1 & 0 & \cdots & n & 0 \\ -1 & 0 & \cdots & 0 & n \end{vmatrix}.$$

( $C_j - C_1$ , puis  $L_1 + L_j$  pour tout  $j > 1$ )

On a alors  $n-2$  fois  $n$  sur la diagonale, ce qui nous permet donc de retrouver  $t(K_n) = T_n = n^{n-2}$ .

## Double dénombrement

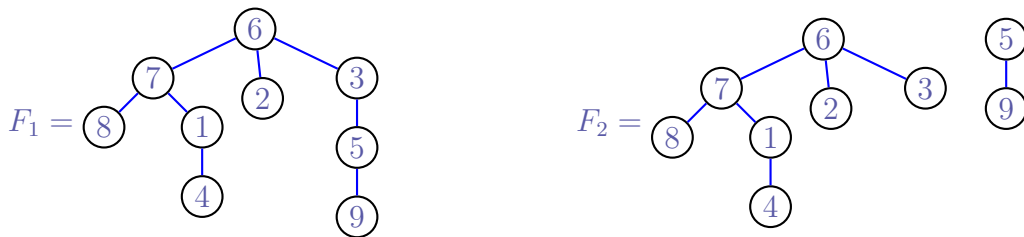
Une seconde manière très élégante, la plus élégante d'après Martin Aigner et Günter M. Ziegler dans *Proofs from THE BOOK*, est donnée par Jim Pitman [Pit99]. L'auteur utilise la technique de double dénombrement pour retrouver la formule de Cayley. Celle-ci s'intéresse non pas aux arbres, mais bien aux forêts d'arbres enracinés.

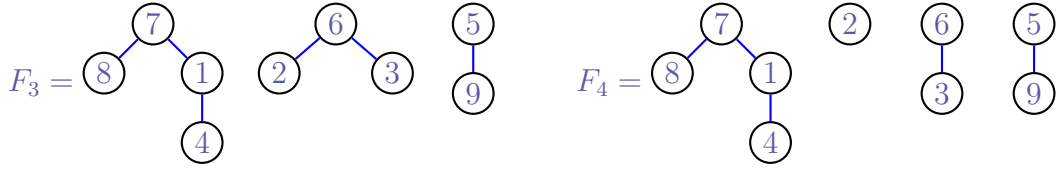
On pose  $\mathcal{F}_{n,k}$  l'ensemble des forêts d'arbres enracinés à  $n$  nœuds avec  $k$  racines.

**Remarque 5.2.3 :** Pour  $k = 1$ , le nombre de forêts d'arbres enracinés ayant une racine,  $\#\mathcal{F}_{n,1}$ , correspond au nombre d'arbres enracinés  $\#\mathcal{A}[n]$  à  $n$  nœuds (cf. Équation 1.19). De plus, ce nombre est  $n$  fois le nombre d'arbres (non-enracinés) à  $n$  nœuds,  $nT_n$  (et  $n$  correspond au nombre de manières de choisir une racine).

On appelle *séquence raffinée*  $F_1, \dots, F_k$  une séquence de  $k$  forêts d'arbres enracinés tels que  $F_i \in \mathcal{F}_{n,i}$  et tels que  $F_{i+1}$  est (en tant que graphe) un sous-graphe de  $F_i$ .

**Exemple 5.2.4 :** La séquence de forêts suivante satisfait ces contraintes :





C'est une *séquence raffinée*.

**Lemme 5.2.5 ([Pit99, Lemma 1]):** Soit  $F_k \in \mathcal{F}_{n,k}$  une forêt de  $k$  arbres enracinés fixée sur l'ensemble  $U$  à  $n$  éléments. Le nombre d'arbres enracinés (sur l'ensemble  $U$ ) qui contiennent  $F_k$  est  $n^{k-1}$ .

PREUVE: On note  $N(F_k)$  le nombre d'arbres enracinés sur  $U$  qui contiennent  $F_k$  (le nombre d'arbres/forêts  $F_1$  possibles dans les séquences raffinées). On pose  $N^*(F_k)$  le nombre de *séquences raffinées*  $(F_1, F_2, \dots, F_k)$  possibles.

Toutes les séquences raffinées  $(F_1, \dots, F_k)$ , avec  $F_1$  et  $F_k$  fixés, correspondent à toutes les manières de choisir l'ordre des  $k-1$  arêtes « manquantes » de  $F_k$  qui sont dans  $F_1$ . C'est-à-dire, on a  $k-1$  manières de choisir  $F_2$ , puis  $k-2$  manières de choisir  $F_3$  et ainsi de suite jusqu'à  $F_k$ . On déduit donc l'égalité :

$$N^*(F_k) = N(F_k) (k-1)! . \quad (5.4)$$

Par ailleurs, partant de  $F_k$ , le nombre de manières de construire  $F_{k-1}$  correspond au nombre de manières de choisir un nœud  $u \in U$  et une des  $k$  racines  $r$  tels que  $u$  et  $r$  n'appartiennent pas à la même composante connexe dans  $F_k$ . Ce nombre est  $n(k-1)$ , c'est-à-dire le choix d'un nœud  $u$  fois le choix d'une des  $k-1$  racine  $r$  des arbres qui ne contiennent pas  $u$ . Pour tout  $k \geq j \geq 2$ , le nombre de manières de construire  $F_{j-1}$  à partir de  $F_j$  est  $n(j-1)$ . On déduit alors que le nombre de manières de construire une séquence raffinée à partir de  $F_k$  est donc

$$N^*(F_k) = n^{k-1} (k-1)! . \quad (5.5)$$

À partir de (5.4) et (5.5), on déduit donc :

$$N(F_k) = n^{k-1} . \quad \blacksquare$$

Ce lemme a pour conséquence que  $N(F_n) = \#\mathcal{F}_{n,1} = n^{n-1}$  et d'après la Remarque 5.2.3, on retrouve la formule de Caley :  $T_n = n^{n-2}$ .

### 5.2.2. Codage de Prüfer

On peut aussi citer la preuve de Heinz PRÜFER, en 1918, qui met en bijection les mots de longueur  $n-2$  sur un alphabet de taille  $n$  (qui sont bien évidemment au nombre de  $n^{n-2}$ ) et les arbres. Cette bijection est connue comme le *codage de Prüfer* [Prü18]. Dans les deux premiers paragraphes de cette sous-section, on décrit ce codage au travers des Algorithmes 5.2.1 et 5.2.2 (sans donner de preuve qu'il s'agit d'une bijection).

Ce codage permet d'introduire la bijection de Henri O. POLLAK entre les fonctions de parking et les *séquences de Prüfer*, et ainsi les arbres et forêts d'arbres enracinés. Cette bijection présentée dans [Rio69] par John RIORDAN expose une bijection élégante, faisant apparaître ce codage compact des arbres. Cette dernière a été démontrée dans [FR74] par Dominique FOATA et John RIORDAN.

## Codage

Dans ce premier paragraphe, on donne un algorithme associant, à tout arbre à  $n$  nœuds, un unique mot de longueur  $n - 2$  sur l'ensemble  $[n]$ , c'est-à-dire une *séquence de Prüfer*.

Soit  $A$  un arbre sur l'ensemble  $[n]$ . On définit l'ensemble des *feuilles* de  $A$  comme l'ensemble des nœuds n'ayant qu'un arc sortant, c'est-à-dire l'ensemble des nœuds n'ayant qu'un unique nœud adjacent. On définit aussi le *parent* d'une feuille  $q$  comme l'unique nœud  $p$  adjacent à  $q$ .

Premièrement, l'algorithme qui associe, à tout arbre, une séquence de Prüfer :

---

**Algorithme 5.2.1** : Arbre vers séquence de Prüfer

---

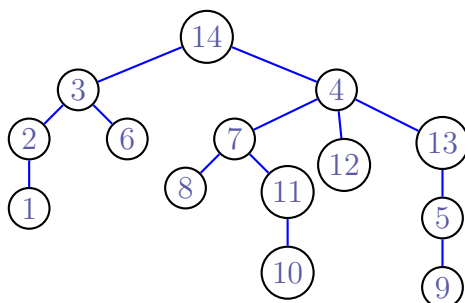
**Données** : un arbre  $A$  sur l'ensemble  $[n]$

**Résultat** : une séquence de Prüfer  $S = (p_i)$  de  $n - 2$  entiers

```
1  $S \leftarrow ()$ ; // séquence vide d'entiers
2 tant que  $|A| > 2$  faire
3    $m \leftarrow \min\{q \text{ où } q \text{ est une feuille de } A\}$ ;
4    $p \leftarrow$  le parent de  $m$ ;
5    $S \leftarrow S \cdot p$ ;
6   supprimer  $m$  (ainsi que l'arc  $\{m, p\}$ ) de  $A$ ;
7 retourner  $S$ 
```

---

**Exemple 5.2.6** : On reprend la forêt à 13 nœuds de l'Exemple 5.2.1 précédent. L'égalité  $\mathcal{F} = \mathcal{A}'$  permet d'associer (bijectivement) à cette forêt à l'arbre (non-enraciné) :



où le nœud d'étiquette 14 peut-être considéré comme une racine virtuelle de la forêt (pour plus de détail sur cette construction, voir [BLL98, § 2.2.2].)

L'Algorithme 5.2.1 construit la séquence  $(2, 3, 3, 14, 7, 5, 13, 11, 7, 4, 4, 4)$ .

## Décodage

Pour reconstruire l'arbre à partir d'une séquence de Prüfer (autrement dit de n'importe quel mot de  $[n]$  de longueur de  $n - 2$ ) :

---

**Algorithme 5.2.2 :** Séquence de Prüfer vers arbre

---

**Données :** une séquence de Prüfer  $S = (p_i)$  de  $n - 2$  entiers (sur  $[n]$ )

**Résultat :** un arbre  $A$  sur l'ensemble  $[n]$

---

- 1  $A \leftarrow$  le graphe sans arcs sur l'ensemble  $[n]$  ;
  - 2  $D \leftarrow (d_i)$  avec  $d_i :=$  nombre d'occurrences de  $i$  dans  $S$ , pour  $1 \leq i \leq n - 2$  ;
  - 3 **pour tous les**  $1 \leq i \leq n - 1$  **faire**
  - 4      $j \leftarrow \min\{j \mid d_j = 0\}$  ;
  - 5     ajouter l'arc  $\{p_i, j\}$  dans  $A$  ;
  - 6     décrémenter  $d_{p_i}$  et  $d_j$  dans  $D$  ;
  - 7 ajouter un arc entre les deux derniers  $j$  tels que  $d_j = 0$  ;
  - 8 **retourner**  $A$
- 

**Exemple 5.2.7 :** On reconstruit l'arbre associé à la séquence de Prüfer  $(1, 7, 5, 7, 7, 1)$  à partir de l'Algorithme 5.2.2 :

–  $A$  :



le graphe ayant  $[8]$  comme ensemble de sommet et aucune arête.

–  $D := (2, 0, 0, 0, 1, 0, 3, 0)$  ;

–  $i = 1 : (d_1 = 1)$

–  $j = 2$ ,

–  $A$  :



–  $D := (1, -1, 0, 0, 1, 0, 3, 0)$  ;

–  $i = 2 : (d_2 = 7)$

–  $j = 3$

–  $A$  :



–  $D := (1, -1, -1, 0, 1, 0, 2, 0)$  ;

–  $i = 3 : (d_3 = 5)$

–  $j = 4$ ,

–  $A$  :



–  $D := (1, -1, -1, -1, 0, 0, 2, 0)$  ;

–  $i = 4 : (d_4 = 7)$

–  $j = 5$ ,

–  $A$  :

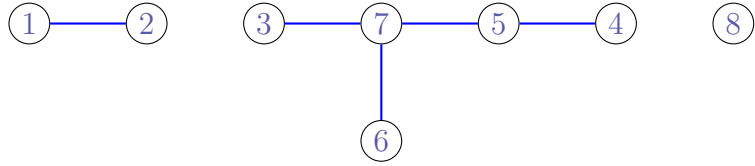


–  $D := (1, -1, -1, -1, -1, 0, 1, 0)$  ;

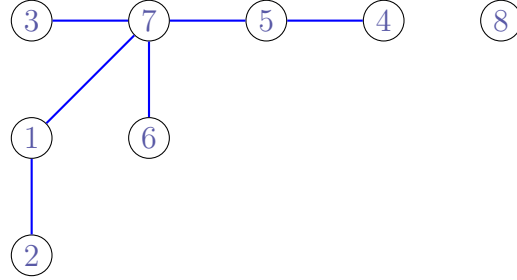
–  $i = 5 : (d_5 = 7)$

–  $j = 6$ ,

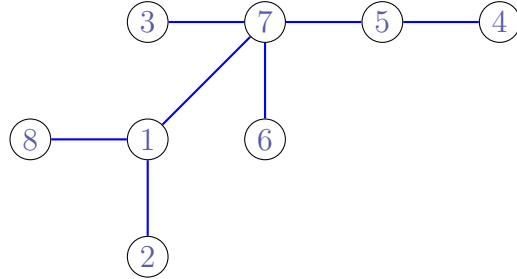
–  $A$  :



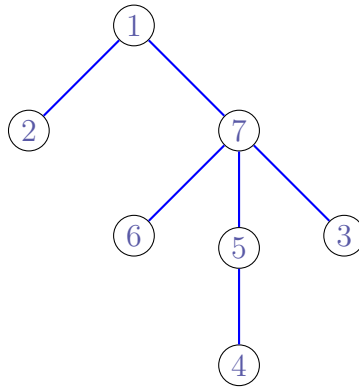
- $D := (1, -1, -1, -1, -1, -1, 0, 0)$ ;
- $i = 6 : (d_6 = 1)$
- $j = 7$ ,
- $A$  :



- $D := (0, -1, -1, -1, -1, -1, -1, 0)$ ;
- et enfin  $A$  :



Cet arbre (non-enraciné) correspond dans notre cadre des forêts d'arbres enracinés à la forêt ne contenant qu'un seul arbre enraciné de racine 1 (où le sommet 8 est la racine virtuelle), c'est-à-dire :



## Bijection

Soit  $f : [n] \rightarrow \mathbb{N}_{>0}$  une fonction de parking sur l'ensemble  $[n]$ , définie comme une fonction entière (cf. § 3.3). La construction de Henri O. POLLACK construit la séquence



de Prüfer (de longueur  $n - 1$  sur l'ensemble  $[n + 1]$ ) :

$$S = (p_i) \quad \text{avec} \quad p_i = f(i + 1) - f(i) \mod (n + 1) .$$

Cette bijection discutée dans [Rio69] et a été prouvée dans [FR74]. À partir d'une telle bijection, on établit alors que la série génératrice des fonctions de parking satisfait l'équation fonctionnelle :

$$\mathcal{P}(t) = \exp(t \mathcal{P}(t)) .$$

Cette équation découle immédiatement de la définition de l'espèce des forêts d'arbres enracinés (1.20).

### 5.3. Bijection FOATA-RIORDAN-YAN

Dans [FR74], les auteurs proposent une seconde bijection basée sur des ordres biens choisis de l'ensemble d'étiquettes  $U$ . Cette idée a été reprise par Catherine H. YAN dans [Yan00 ; Yan01] pour établir une bijection plus générale entre les  $(a(m - 1) + b)$ -fonctions de parking et les structures arborescentes de l'espèce des  $a, b$ -arborescences :

$$\mathcal{F}^{a,b} := E(X \cdot \mathcal{F}^{a,a})^b , \quad (5.6)$$

avec  $a, b \geq 0$ . Dans cette section, on généralise, encore, très naturellement ces bijections aux  $(a(m - 1) + b)$ -tables de parking en  $\mathbf{F}$  et aux structures arborescentes définies par l'espèce des  $a, b$ -arborescences en  $\mathbf{F}$  :

$$\mathcal{F}_F^{a,b} := F(X \cdot \mathcal{F}_F^{a,a})^b . \quad (5.7)$$

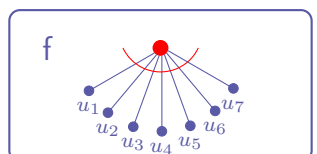
Dans notre contexte de tables de parking définies en l'espèce  $\mathbf{F}$  et d'arborescences  $\mathcal{F}_F^{a,b}$ , les notions d'ordres bien choisis apparaissent comme des lectures « naturelles » des structures. Dans le cas des tables de parking, il s'agit d'une lecture de gauche à droite des étiquettes ; et, dans le cas des arborescences, cet ordre correspond à une lecture en largeur.

#### 5.3.1. Structures arborescentes

Pour faciliter (et éviter d'être verbeux inutilement), on choisit de représenter de manière abstraite les différentes structures par des schémas (cf. [BLL98]).

##### $\mathbf{F}$ -structures

On représente une  $\mathbf{F}$ -structure  $f$  sur  $U$  par le schéma :



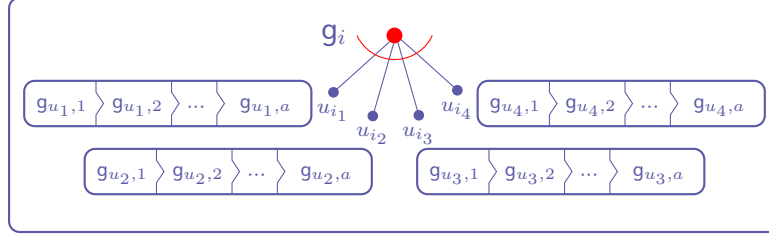
Les étiquettes de l'ensemble  $U$  sont représentées sur ce schéma par les points  $u_i$ .

### $\mathcal{F}_F^{a,b}$ -structures

Une  $\mathcal{F}_F^{a,b}$ -structure sur  $U$  comme une séquence  $(g_i)$  représentée par le schéma :



où chaque structure  $g_i$  se représente par le schéma (récursif) :



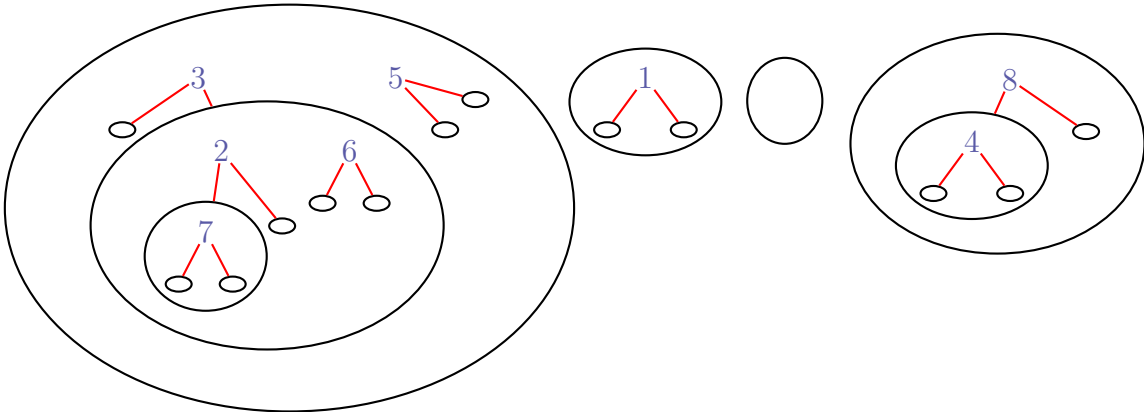
où chaque étiquette  $u_j$  (correspondant à une structure de singleton) forme un couple avec une séquence de  $\mathcal{F}_F^{a,a}$ -structures :  $(u_j, (g_{u_j,1}, \dots, g_{u_j,a}))$ .

**Remarque 5.3.1 :** Formellement, la définition de l'opération de composition des espèces définit ces structures comme un triplet  $(\lambda, s, t)$  où  $\lambda$  est une partition de  $U$ ,  $s$  est une  $F$ -structure sur  $\lambda$  et  $t$  est un ensemble où chaque élément est une  $(X \cdot \mathcal{F}_F^{a,a})$ -structure définie pour chaque ensemble de part de  $\lambda$ . L'unique appareillage des étiquettes/parts  $p$  de  $\lambda$  dans  $s$  avec la  $X \cdot \mathcal{F}_F^{a,a}$ -structure sur  $p$  encode exactement et sans ambiguïté à la description précédente.


Dans la suite, on associe à chaque  $g_j$  une indexation correspondant à une lecture en largeur de l'arborescence. Pour se faire, on donne quelques définitions sur ces structures : Soit  $(u_j, (g_{j_1}, \dots, g_{j_a}))$  un tuple-étiquette d'une arborescence  $g$  d'une  $\mathcal{F}_F^{a,b}$ -structure.

- la structure  $g$  est le *parent* de l'étiquette  $u_j$  et de chaque  $\mathcal{F}_F^{a,a}$ -structure  $g_{j_i}$ , respectivement notés  $p(u_j)$  et  $p(g_{j_i})$  ;
- chaque structure  $g_{j_i}$  (pour  $1 \leq i \leq a$ ) est *rattachée à l'étiquette*  $u_j$ , noté  $e(g_{j_i})$  ;
- une structure  $g$  est une *racine* si elle n'est rattachée à aucune étiquette ;
- la *hauteur* d'une étiquette  $u \in U$ , noté  $h(u)$ , est définie par  $h(u) = 0$  si  $p(u)$  est une racine et  $h(e(p(u))) + 1$  sinon.

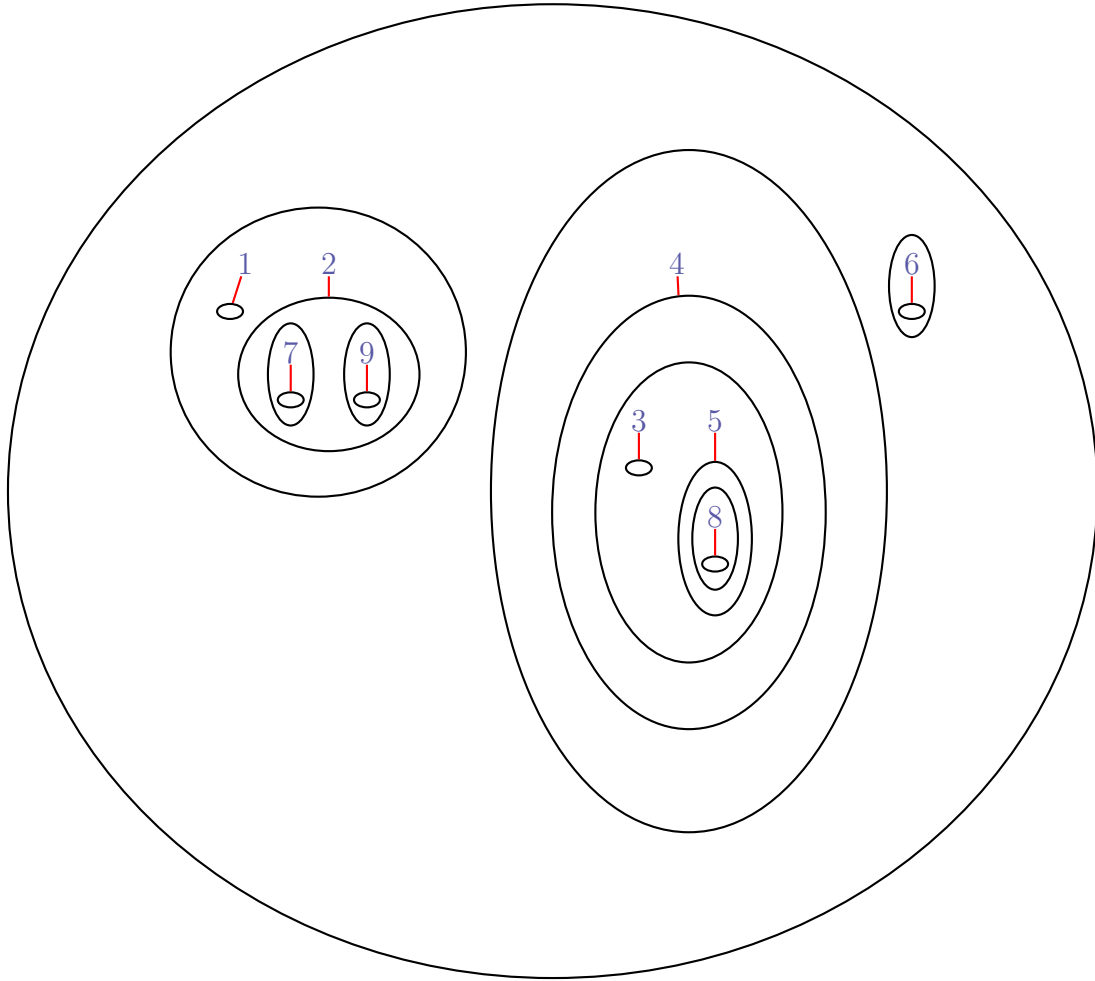
**Exemple 5.3.2 :** Voici une structure arborescente de  $\mathcal{F}_E^{2,4} = \mathcal{F}^{2,4}$  sur l'ensemble [8] :



où la première forêt (à gauche) est une  $\mathcal{F}^{2,2}$ -structure sur l'ensemble  $\{2, 3, 5, 6, 7\}$  qui est composée de deux arbres respectivement enracinés en 3 et 5 ; la seconde est une  $\mathcal{F}^{2,2}$ -structure sur l'ensemble  $\{1\}$ , la troisième est la forêt vide, et la quatrième est une  $\mathcal{F}^{2,2}$ -structure sur l'ensemble  $\{4, 8\}$  qui est composée d'un arbre enraciné en 8.

Les nœuds 2 et 6 ont pour *parent* le nœud 3 et sont à *hauteur* 1. Le nœud 8 est la *racine* des forêts  (à gauche) et celle vide (à droite).

**Exemple 5.3.3 :** Soit *Par* l'espèce des partitions (1.21). Une  $\mathcal{F}_{\text{Par}}^{1,1}$ -structure (ou hyper-arbre (couvrant) au sens de [A030019]) sur [9] :



**Proposition 5.3.4 :** Soit  $(g_i)$  une  $\mathcal{F}_{\mathbb{F}}^{a,b}$ -structure sur l'ensemble  $U$  à  $n$  éléments.

Le  $b$ -uplet  $(g_i)$  se décompose exactement en  $an + b$  structures distinctes de l'espèce  $\mathcal{F}_{\mathbb{F}}^{a,a}$ .

PREUVE : Si  $n = 0$  alors  $(g_i)$  contient bien  $b$  forêts vides. Sinon, par définition, chaque étiquette est rattachée à  $a$  structures distinctes de l'espèce  $\mathcal{F}_{\mathbb{F}}^{a,a}$ . D'où  $U$  un ensemble de  $n$  étiquettes, une  $\mathcal{F}_{\mathbb{F}}^{a,b}$ -structure  $(g_i)$  est bien constituée de  $b + an$  structures (par définition distinctes) de l'espèce  $\mathcal{F}_{\mathbb{F}}^{a,a}$ . ■

### 5.3.2. Arborescences vers tables de parking

On commence par généraliser la bijection de FOATA-RIORDAN-YAN en définissant une première construction de  $(a(m-1) + b)$ -tables de parking en  $\mathbb{F}$  à partir de  $\mathcal{F}_F^{a,b}$ .

Pour la suite, on se fixe  $\ell \ll U$  un ordre total sur l'ensemble fini  $U$  (de  $n$  étiquettes). Cet ordre total va nous permettre d'associer à chaque arborescente (et par la suite à chaque table de parking) un nouvel ordre  $<_q$ .

Soit  $(g_i)$  une  $\mathcal{F}_F^{a,b}$ -structure sur  $U$ .

#### Lecture en largeur des étiquettes

On définit l'ordre  $<_q$  sur  $U$  par une lecture en largeur de l'arborescence :

$$u <_q v \iff \begin{cases} u <_\ell v & \text{et } p(u) = p(v), \\ i < j & \text{avec } p(u) = g_i \text{ et } p(v) = g_j \text{ deux racines,} \\ e(p(u)) <_q e(p(v)) & \text{et } h(u) = h(v), \\ h(u) < h(v) & \text{sinon.} \end{cases}$$

**Exemple 5.3.5 :** L'ordre  $<_q$  sur  $[8]$  associé au tuple de forêt de l'Exemple 5.3.2 est :

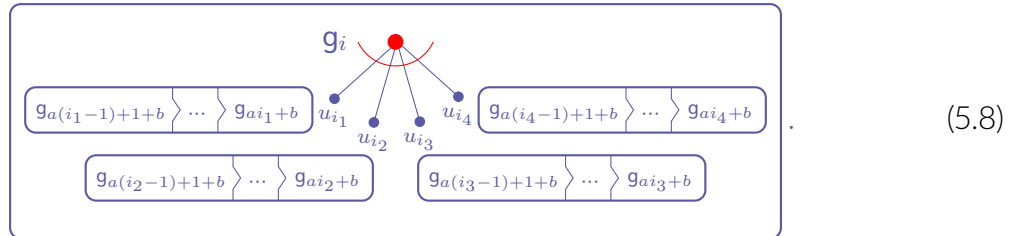
$$3 <_q 5 <_q 1 <_q 8 <_q 2 <_q 6 <_q 4 <_q 7.$$

**Exemple 5.3.6 :** L'ordre obtenu à partir de l'hyper-arbre de l'Exemple 5.3.3 est

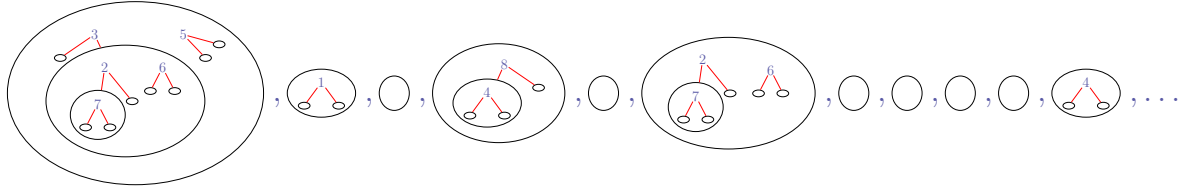
$$1 <_q 2 <_q 4 <_q 6 <_q 7 <_q 9 <_q 3 <_q 5 <_q 8.$$

#### Lecture en largeur des $\mathcal{F}_F^{a,b}$ -structures

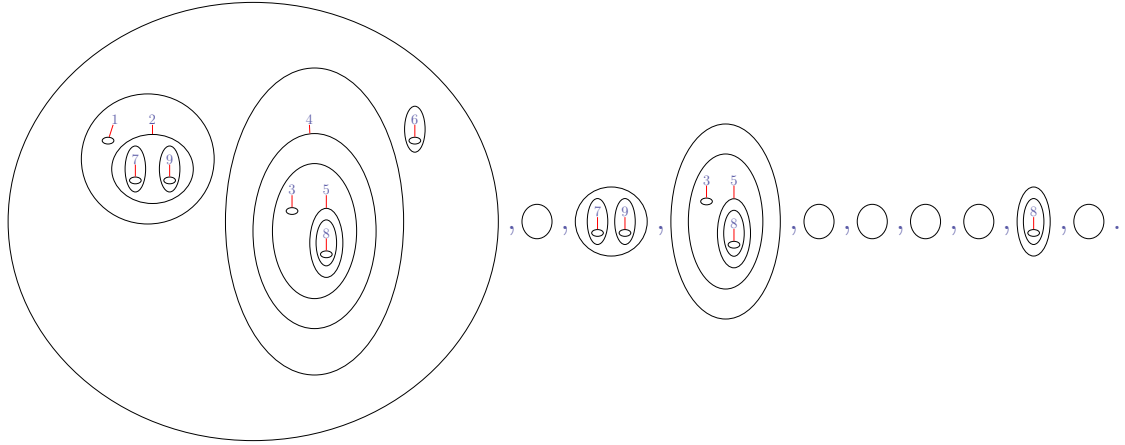
On attribut à chaque structure  $g$  qui apparaît (récursivement) dans l'arborescence  $(g_i)$  un index en fonction de  $<_q$  et de sa position dans le tuple qui lui est associé : Si  $g$  est une *racine* alors son index correspond à sa position  $1 \leq i \leq b$  dans le tuple  $(g_i)$  ; sinon  $g$ , en  $j^{\text{ième}}$  position dans un  $a$ -uplet, est *rattachée* à l'étiquette  $u_i$ , en  $i^{\text{ième}}$  position pour l'ordre  $<_q$ , on lui associe alors l'unique index  $a(i-1) + j + b$  :



**Exemple 5.3.7 :** Les indexations des deux précédentes structures (des Exemples 5.3.2 et 5.3.3) construisent respectivement les ordres :



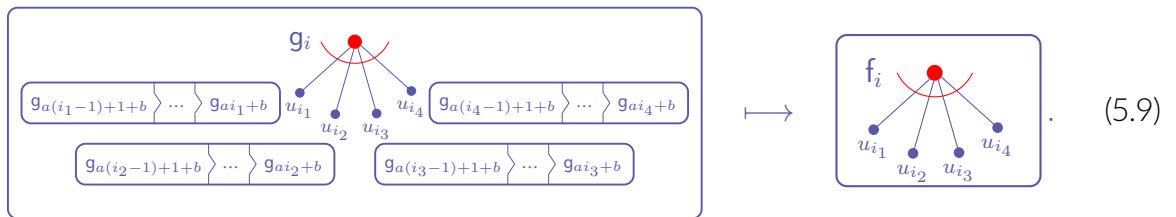
et



À partir de cette indexation, on va construire naturellement la séquence (associée à cet ordre) des  $\mathbf{F}$ -structures sous-jacentes qui est une  $(a(m-1)+b)$ -table de parking en  $\mathbf{F}$  :

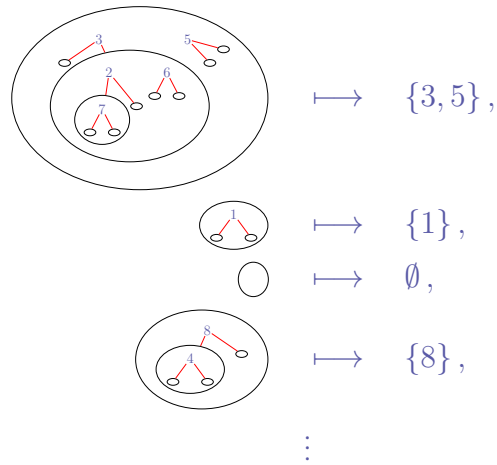
### Construction : tables de parking

À chaque  $\mathcal{F}_F^{a,a}$ -structure  $\mathbf{g}_i$ , on peut naturellement associer  $\mathbf{f}_i$  la  $\mathbf{F}$ -structure sous-jacente à  $\mathbf{g}_i$ , en oubliant les  $a$ -uplets de  $\mathcal{F}_F^{a,a}$ -structures rattachés à chaque étiquette  $u \in U$  :

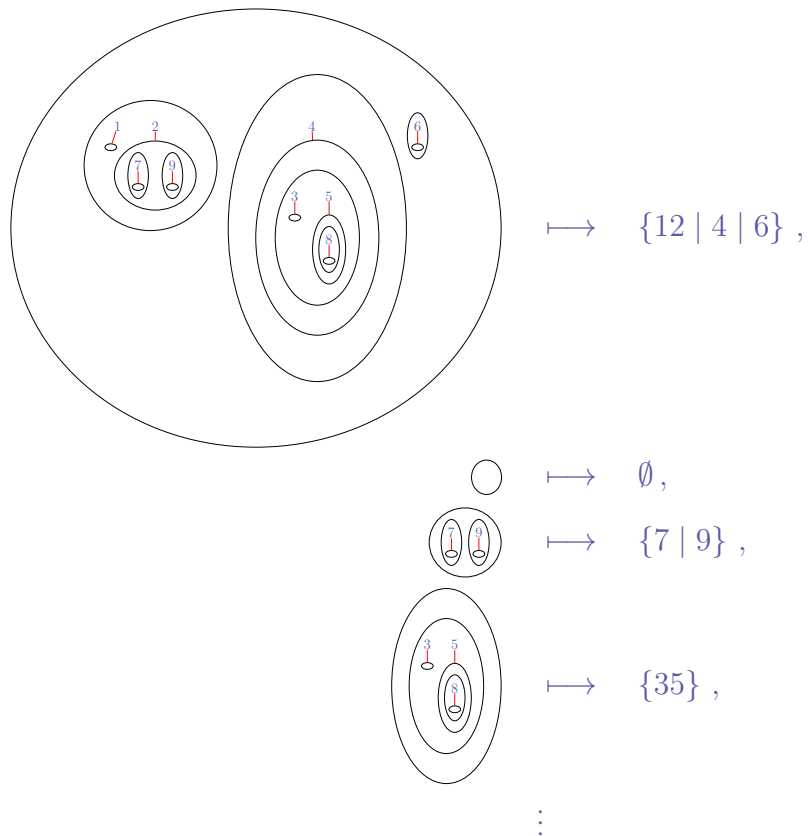


**Remarque 5.3.8 :** Formellement, il faudrait considérer le triplet  $(\lambda, s, t)$  et extraire, de chaque part de  $\lambda$  dans  $s$ , les étiquettes autres que celles qui sont les *étiquettes rattachées* dans  $t$ .

**Exemple 5.3.9 :** En reprenant les sous  $\mathcal{F}^{5,2}$ -structures présentées dans l'Exemple 5.3.7 produisent les ensembles suivants :



Et de même, les sous  $\mathcal{F}_{\text{Par}}^{1,1}$ -structures présentées dans l'Exemple 5.3.3 produisent les partitions suivantes :



(cf. notation des partitions d'ensembles § 1.1.2.)

En utilisant l'indexation de chaque  $\mathbf{g}$ , on construit alors une séquence de  $a\mathbf{n} + b$   $\mathbf{F}$ -structures  $(\mathbf{f}_i)$  qui est une  $(a(m-1) + b)$ -table de parking en  $\mathbf{F}$  :

**Lemme 5.3.10 :** La construction est une application injective de  $\mathcal{F}_F^{a,b}[U]$  dans  $\triangleright_{a(m-1)+b}^F[U]$ .

PREUVE :

**Injectivité :** La Proposition 5.3.4 nous assure que  $(g_i)$  est un bien une séquence de longueur  $an + b$ , composée de  $\mathcal{F}_F^{a,b}$ -structures distinctes. D'après la construction (5.9), à partir de chaque structure  $g_i$ , on produit une  $F$ -structure  $f_i$  en émondant  $g_i$  au niveau de ces étiquettes  $u_k$  de hauteur minimale (c'est-à-dire à hauteur 0 par rapport à la structure  $g_i$ ). Récursivement, chaque structure  $g_{a(k-1)+j+b}$  rattachée à chaque étiquette  $u_k$  produit une  $F$ -structure  $f_{a(k-1)+j+b}$ . L'indexation des structures induit que cette construction est réversible (*modulo* la préservation de l'ordre  $<_q$ ) et est donc injective.

**Contrainte de parking :** Si  $n = 0$ , la contrainte de parking (3.1) est nécessairement satisfaite, on obtient une séquence  $(g_i)$  de  $b$  arborescences vides. On suppose cette condition satisfaite jusqu'à un certain rang  $n - 1$ . Soit  $U$  (à  $n$  éléments), par hypothèse, la séquence  $(f_i)$  de  $F$ -structures (de longueur au moins  $an + b$ ) construite à partir des  $n - 1$  premiers éléments de  $U$  (suivant l'ordre  $<_q$ ) satisfait la contrainte, c'est-à-dire

$$\sum_{i=1}^{a(k-1)+b} \#Q_i \geq k, \quad \text{pour tout } k < n,$$

avec  $Q_i$  l'ensemble sous-jacent à  $f_i$  (c'est-à-dire l'ensemble tel que  $f_i \in F[U_i]$ ). Par définition de la lecture en largeur des  $\mathcal{F}_F^{a,b}$ -structures, la dernière étiquette de  $U$  est rencontrée au plus tard à l'avant-dernier élément de  $U$  (pour l'ordre  $<_q$ ). Par définition de l'indexation (5.8), cette étiquette est rencontrée au plus tard lors de la lecture d'une structure  $g_j$  avec  $a(n - 1) + b < j < an + b$ . Par construction de la séquence  $(f_i)$ , cela nous assure que

$$\sum_{i=1}^{a(k-1)+b} \#Q_i \geq k, \quad \text{pour tout } k \leq n.$$

Par récurrence, on déduit donc que  $(f_i)$  est une  $(a(m - 1) + b)$ -table de parking en  $F$  sur  $U$ . ■

**Exemple 5.3.11 :** En reprenant le tuple de forêts de l'Exemple 5.3.2, cette construction définit la  $(2m + 2)$ -fonction de parking sur [8] (note :  $2m + 2 = 2(m - 1) + 4$ ) :

$$(35 \mid 1 \mid \cdot \mid 8 \mid \cdot \mid 26 \mid \cdot \mid \cdot \mid \cdot \mid \cdot \mid 4 \mid \cdot \mid 7 \mid \cdot \mid \cdot \mid \cdot \mid \cdot \mid \cdot \mid \cdot \mid \cdot).$$

**Exemple 5.3.12 :** L'exemple 5.3.3 est en bijection avec la Id-table de parking en  $\text{Par}$  :

$$(\{12 \mid 4 \mid 6\}, \emptyset, \{7 \mid 9\}, \{35\}, \emptyset, \emptyset, \emptyset, \emptyset, \{8\}, \emptyset).$$

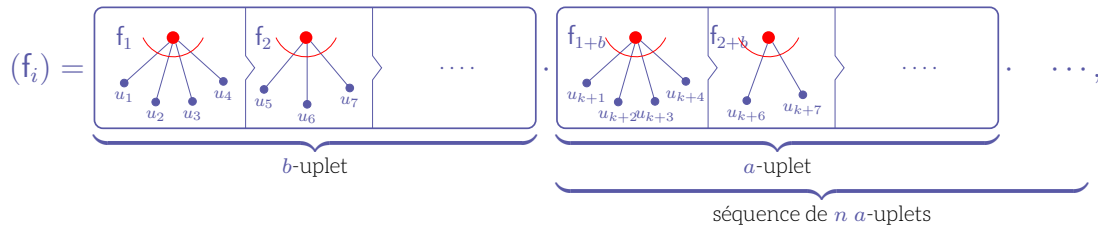
### 5.3.3. Tables de parking vers arborescences

Dans la précédente sous-section, on a vu que la construction des  $(a(m-1)+b)$ -tables de parking en  $\mathbf{F}$  à partir d'une structure arborescente de  $\mathcal{F}^{a,b}$  est réversible. Dans cette sous-section, on montre que l'inverse existe quelque soit la  $(a(m-1)+b)$ -table de parking considérée.

Soit  $(f_i)$  une  $(a(m-1)+b)$ -table de parking en  $\mathbf{F}$  sur  $U$ .

#### $(a(m-1)+b)$ -tables de parking en $\mathbf{F}$

On représente  $(f_i)$  par le schéma :



où on rappelle que le dernier  $a$ -uplet de  $\mathbf{F}$ -structure est nécessairement défini sur l'ensemble vide.

#### Lecture de gauche à droite

Chaque  $f_i$  est une  $\mathbf{F}$ -structure sur un sous-ensemble (disjoint)  $U_i$  de  $U$ . On construit un nouvel ordre  $<_q$  sur  $U$  :

$$u_1 <_q u_2 <_q \cdots <_q u_n,$$

en lisant la table de parking de gauche à droite :

$$u <_q v \iff \begin{cases} u <_\ell v & \text{et } u, v \in U_i, \\ u \in U_i, v \in U_j & \text{et } i < j; \end{cases} \quad (5.10)$$

où  $u, v \in U_i$  signifie que  $u$  et  $v$  sont deux étiquettes de la  $i^{\text{ième}}$ -structure  $g_i$  telle que  $g_i \in \mathbf{F}[U_i]$ .

**Exemple 5.3.13 :** Soit  $(\cdot \mid 23 \mid \cdot \mid \cdot \mid \cdot \mid \cdot \mid \cdot \mid \cdot \mid \cdot \mid 1 \mid \cdot \mid \cdot \mid \cdot \mid \cdot)$  une  $(3m+2)$ -table de parking en  $\mathbf{E}$  (ou fonction de parking) sur l'ensemble  $[3]$  (naturellement ordonné). L'ordre total  $<_q$  correspond à la lecture de gauche à droite de la fonction de parking suivant l'ordre  $\ell$  :

$$2 <_q 3 <_q 1.$$

**Exemple 5.3.14 :** Soit  $\mathcal{C}$  l'espèce des cycles (ex.  $(123) = (231) = (312)$ ) [BLL98, § 1.1]. La  $2m$ -table de parking en  $(\mathcal{C} + 1)$  (l'espèce des cycles incluant le cycle vide, noté  $()$ ) sur [8] :

$$(), (264), (), (), (18), (), (), (), (), (), (7), (35), (), (), (), (), ()$$



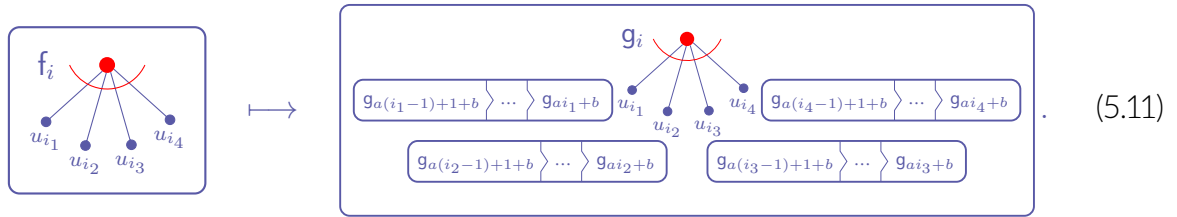
produit l'ordre suivant

$$2 <_q 4 <_q 6 <_q 1 <_q 8 <_q 7 <_q 3 <_q 5.$$

(**Remarque :** sans perdre de généralité, on pourrait se fixer un autre ordre en utilisant un ordre sur les cycles, par exemple  $2 <_q 6 <_q 4 <_q \dots$ .)

### Construction : arborescence

On construit récursivement une  $\mathcal{F}_F^{a,b}$ -structure  $(g_i)_{i \in [b]}$  sur  $U$  à partir de  $(f_i)$  de la manière suivante : pour tout  $1 \leq i \leq an + b$ , on substitue dans  $f_i$  chaque étiquette  $u_j$  par le couple  $(u_j, (g_{a(j-1)+1+b}, \dots, g_{aj+b}))$  où chaque  $g_k$  est récursivement défini :



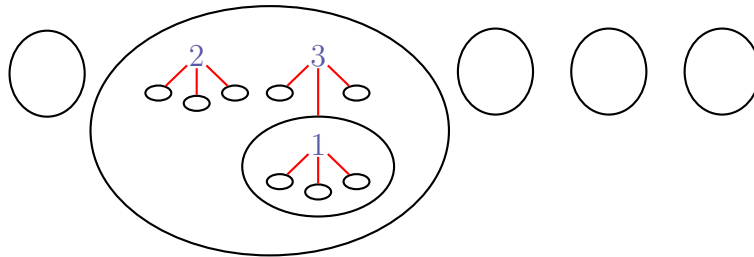
**Exemple 5.3.15 :** À partir de la  $(3m + 2)$ -fonction de parking de l'Exemple 5.3.13 précédent (note :  $3m + 2 = 3(m - 1) + 5$ ). On construit le 5-uplet de 3-forêts :  $(F_1, F_2, F_3, F_4, F_5)$  où

$$F_2 = \left\{ \left( 2, T_1^{(1)}, T_2^{(1)} T_3^{(1)} \right), \left( 3, T_1^{(2)}, T_2^{(2)} T_3^{(2)} \right) \right\},$$

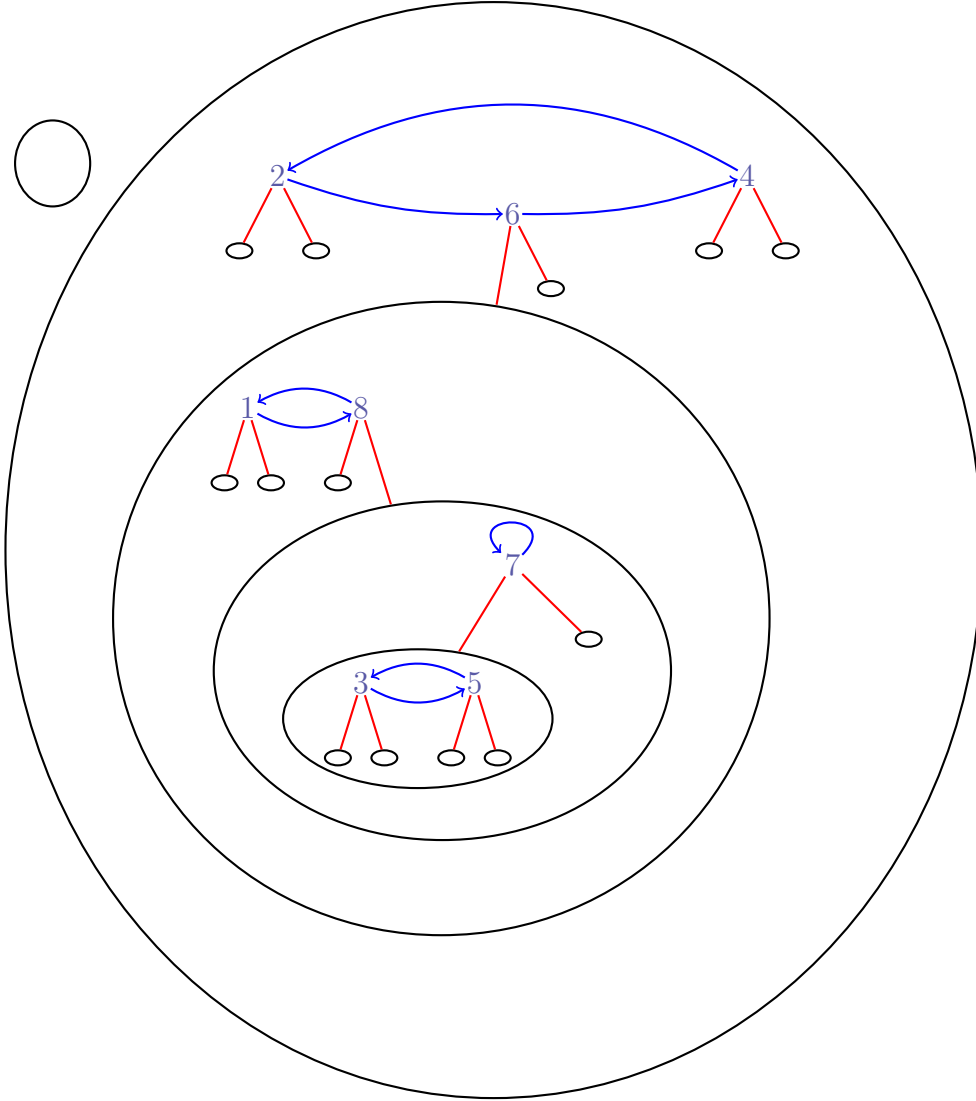
et les autres forêts  $F_1, F_3, F_4, F_5$  sont vides. Les  $T_j^{(1)}$  sont des  $a$ -forêts (ordonnées d'arbres) vides de racine 2, notées  $\emptyset$ ; et,

$$T_1^{(2)} = \emptyset, \quad T_2^{(2)} = \{(1, \emptyset, \emptyset, \emptyset)\}, \quad T_3^{(2)} = \emptyset.$$

Cette  $\mathcal{F}^{3,2}$ -structure sur  $[3]$  se représente alors par :



**Exemple 5.3.16 :** La  $\mathcal{F}_C^{2,2}$ -structure sur [8] construite à partir de l'Exemple 5.3.14 est :



avec les cycles sous-jacents représentés par les arêtes bleus et les arborescentes sous-jacentes reliés aux étiquettes par les arcs rouges (ordonnés).

**Lemme 5.3.17 :** *L construction est une application injective de  $\triangleright_{a(m-1)+b}^F[U]$  dans  $\mathcal{F}_F^{a,b}[U]$ .*

PREUVE :

**Application :** Par récurrence sur le nombre d'étiquettes de  $U$ , si  $n = 0$  on a  $(f_i) \simeq (g_i)$  (car  $F^b[\emptyset] \simeq \mathcal{F}_F^{a,b}[\emptyset]$  d'après la définition de la composition d'espèces et l'hypothèse que  $X \cdot \mathcal{F}^{a,a}$  est de valuation strictement positive). On suppose la construction valable jusqu'à un certain rang  $n - 1$ . Soit  $U$  un ensemble de  $n$  étiquettes. On peut appliquer la construction d'une  $\mathcal{F}_F^{a,b}$ -structure, à partir d'une  $(a(m-1) + b)$ -table de parking en  $F$  sur  $U$ , sur les  $n - 1$  premières étiquettes de  $U$  (suivant l'ordre  $<_q$ ). D'après la contrainte de parking (3.1), la dernière étiquette  $u$  appartient nécessairement à une  $F$ -structure  $f_i$

pour  $i \leq a(n-1) + b$ ; et, pour tout  $a(n-1) + b < j \leq an + b$ , les structures  $f_j$  sont nécessairement sur l'ensemble vide. La construction associe alors, à cette étiquette  $u$ , la séquence de longueur  $a$  de  $\mathcal{F}_F^{a,a}$ -structures toutes sur l'ensemble vide. Cette construction définit donc bien une application des  $(a(m-1) + b)$ -tables de parking en  $F$  vers les  $\mathcal{F}_F^{a,b}$ -structures tout deux sur l'ensemble  $U$ .

**Injectivité :** Supposons  $(f_i)$  et  $(f'_i)$  deux  $(a(m-1) + b)$ -tables de parking en  $F$  sur  $U$ , distinctes. Soit  $d$  le plus petit indice tel que  $f_d \neq f'_d$ . Par définition de la construction, il existe donc un indice  $j$  tel que les  $\mathcal{F}_F^{a,b}$ -structures construites sont différentes. On déduit donc que l'application est injective. ■

À partir de ces deux précédents Lemmes, on obtient alors :

**Théorème 5.3.18 :** Ces deux applications, des  $(a(m-1) + b)$ -tables de parking en  $F$  vers les  $\mathcal{F}_F^{a,b}$ -structures, définissent une bijection.

PREUVE: Les Lemmes 5.3.17 et 5.3.10 sont respectivement des applications injectives des séquences de forêts dans les fonctions de parking et réciproquement. De plus, les Équations 5.11 et 5.9 sont évidemment inverses l'une de l'autre. On déduit alors des constructions que les ordres  $<_q$  coïncident. Ces constructions sont donc inverses l'une de l'autre. ■

### 5.3.4. Énumération

À partir de cette bijection entre ces deux structures, on déduit automatiquement que la série génératrice des  $(a(m-1) + b)$ -tables de parking en  $F$  satisfait l'équation fonctionnelle :

$$\triangleright_{a(m-1)+b}^F(t) = F(t \triangleright_{am}^F(t))^b.$$

En effet, à partir de la définition de l'espèce  $\mathcal{F}_F^{a,b}$  (5.7), on déduit automatiquement :

$$\mathcal{F}_F^{a,b}(t) = F(t \mathcal{F}_F^{a,a}(t))^b.$$

**Remarque 5.3.19 :** Soit  $\chi : m \mapsto 2m + 2 (= 2(m-1) + 4)$ . À titre de comparaison, le calcul du nombre de  $\chi$ -tables de parking en l'espèce des cycles (incluant le cycle vide) sur les ensembles à vingt éléments prend plus de sept minutes (complexité en temps  $O(2^{2n})$ ) alors que celui du nombre de structures arborescentes données par l'espèce précédente prend moins d'un cinquième de seconde. En utilisant l'équation fonctionnelle des structures arborescentes, l'énumération des structures sur un ensemble à cinq-cents éléments prend vingt minutes. (Les deux calculs ont été réalisés en python (avec SAGE) sur un processeur INTEL Core i5 cadencé à 2,6GHz.)

```
sage: a, b = 2, 4
sage: Sp = Species()
sage: X, O, C = Sp.singletons(), Sp.one(), Sp.cycles()
```

```

sage: CO = C + O
sage: Ba = Sp.recursive_species( 'Ba' )
sage: Ba.define( CO.composite( X * Ba^a ) )
sage: Bb = CO.composite( X * Ba^a )^b
sage: # Nombre de 2,4-structures arborescentes en C+O sur [20]
sage: %time Bb.egs().coefficient(20)
CPU times: user 204 ms, sys: 10.6 ms, total: 215 ms
Wall time: 215 ms
122831628120697993429500899218944

sage: PT = ParkingTables( lambda m: a * (m - 1) + b, CO )
sage: %time PT.egs().coefficient(20)
CPU times: user 7min 11s, sys: 7.4 s, total: 7min 18s
Wall time: 7min 16s
122831628120697993429500899218944

sage: %time Bb.egs().coefficient(100)
CPU times: user 7.6 s, sys: 216 ms, total: 7.82 s
Wall time: 7.67 s
9478847871370279395696034150803850691364083762181015550323
4984738796398280272430837822576817083458046064067356287981
6883286318381375107995677009478544450084115497641191189275
069543031992817661015250802651058992721766450305433600000

sage: %time Bb.egs().coefficient(500)
CPU times: user 20min 45s, sys: 24.7 s, total: 21min 10s
Wall time: 20min 56s
81907865656995897546601088709209254625041495332403287058703
41210072108648526233475635425000679620413072587855199224167
76736255485703440519333610487246151664413466727888878614821
22516027420076398921754759745822266180393668637428042243925
82031804765482741252012019397791598583552856619453828042675
19701213359769566574257903457288754720235012884366400745044
68121041239305173241252941755263148358967638092155167407575
11959091875244221773542244089245683303552899576336537555075
77273978793960855859691177861135297746465171830704185814421
24720303695749295004818374445712367130406946416915878179597
54308568569790117679623381261874525604610657066642742748175
96353983249472919303998559018006239812925762206172088357423
96952584542543763904805338041111705731126424956852504348759
28606798090467639935271731155138064483531234768739615760916
79804231979998870576476048730078253759311831464366134372726
32398651444364121273723953596247200269377501531376880881861
48995373859741470751513783476119622468166779882198835275525

```

96620199294457455631033413790223215342532287164797210339975  
83708174883420586120817292358252125411104167392266255534856  
17159445151840588316605077288266807537607130750850725579407  
96285097750544404766539158082943949407775898951396415546986  
33136278120104314223938613786822728496230848756766553195095  
15433421673340194219839168340559546034914286498256107653412  
26047432383102840431149510763772973456046367372598582123843  
52360349998871444115617318977769089929713111753629676425921  
4295040000000000000000000000000000

## **Deuxième partie .**

### **Énumération d'automates**

The Encyclopedia Galactica defines a robot as a mechanical apparatus designed to do the work of a man. The marketing division of the Sirius Cybernetics Corporation defines a robot as "Your Plastic Pal Who's Fun to Be With. The Hitchhiker's Guide to the Galaxy defines the marketing division of the Sirius Cybernetic Corporation as "a bunch of mindless jerks who'll be the first against the wall when the revolution comes,

(Douglas ADAMS, *The Hitchhiker's Guide to the Galaxy*)

Après avoir rapidement présenté la théorie des automates et le contexte de la théorie des langages, dans un premier chapitre ([Chapitre 6](#)), on s'intéresse à l'énumération des automates de manière générale ([Chapitre 7](#)). On commence par montrer que l'énumération des automates (sans contraintes) est un problème « simple » puis on présente deux différentes techniques afin de raffiner cette énumération. La première technique présentée est une technique usuelle de théorie des graphes utilisée dans [[Lis69](#) ; [Rob85](#)] (ou encore [[Lis06](#)] dans le cas des automates acycliques). En utilisant une technique de double dénombrement, on déduit une formule d'énumération des *automates accessibles*. La seconde technique est une approche naïve (qui semble pourtant ne pas apparaître dans la littérature) de l'énumération des automates minimaux permettant d'extraire les automates qu'on dira *quasi-distingués*. La combinaison des deux techniques permet d'obtenir une formule d'énumération des automates *accessibles quasi-distingués* donnant une meilleure borne supérieure du nombre d'automates minimaux que dans l'article de [[DKS02](#)]. Ces nouvelles formules apportent un nouvel angle d'attaque au problème encore ouvert de l'énumération des automates minimaux.

Dans le chapitre suivant ([Chapitre 8](#)), on s'intéresse à l'énumération des automates acycliques. À partir des résultats de Valery A. LISKOVETS [[Lis06](#)] et l'article de Joseph P. S. KUNG et Catherine H. YAN [[KY03](#)], on déduit immédiatement, de leurs formules, qu'il existe une bijection entre les fonctions de transition des automates acycliques et une certaine famille de fonctions de parking généralisées (cf. [Chapitre 4](#)). On construit une bijection explicite entre ces deux structures. À partir de celle-ci, on déduit des propriétés sur les langages droits des automates immédiatement à partir des fonctions de parking. Ces propriétés permettent d'extraire les informations nécessaires pour raffiner l'énumération des automates acycliques. On obtient successivement des formules pour l'énumération des automates *co-accessibles*, *distingués* et enfin, en appliquant la technique précédente de théorie des graphes, des automates acycliques *minimaux*. Cette dernière partie est une version longue du résumé étendu [[Pri15b](#)].

## 6. Théorie des langages et automates

La *théorie des automates* est l'étude des outils de calculs abstraits. Cette histoire commence, dans les années trente, avec Alan M. TURING. L'objectif de ce pionnier de l'informatique a été de déterminer précisément ce qui est calculable (par un algorithme) de ce qui ne l'est pas.

Dans les années cinquante, Noam CHOMSKY soutient une thèse en linguistique. Il décrit en termes de *langages* une formalisation de ces notions calculables et incalculables ; et il établit une hiérarchie (imbriquée) de ces différents calculs/langages [Cho56] :

- les *langages rationnels* (*réguliers* ou encore *reconnaissables*) : ce sont les langages obtenus à partir des trois opérations, *union*, *intersection* et *étoile de Kleene* appliquées à des ensembles finis (ou récursivement à des ensembles rationnels). D'après le théorème de Kleene [Kle51], ces langages sont exactement tous les langages reconnaissables par l'ensemble des *automates finis*. Ces langages ont un intérêt particulier en théorie de la compilation et en traitement automatique de texte.
- les *langages algébriques* : ce sont les langages définis par une *grammaire algébrique* ou de manière équivalente par un *automate à pile*. Ces langages forment la base des langages de programmation et des technologies du Web (comme XML).
- les *langages contextuels* : ce sont les langages reconnus par une machine de Turing (non déterministe) à mémoire linéairement bornée. Ces langages complexes interviennent principalement dans la présentation de classes de complexité.
- enfin la gamme la plus large de langages, les *langages récursivement énumérables* : ce sont tous les langages reconnus par une machine de Turing.

Dans ce chapitre, on s'intéresse uniquement à la classe des *langages rationnels*, et, en particulier aux *automates finis déterministes* qui les représentent. Dans cette section, on donne rapidement quelques définitions de théorie des langages : alphabet, langages, automates. À partir de ces définitions, nous introduisons la notion d'automates minimaux et nous nous intéressons plus précisément à l'énumération des automates dans la prochaine section. Pour des informations plus approfondies, le lecteur pourra se référer à [HMU07 ; Car08].

### 6.1. Langages rationnels

Un *alphabet* est un ensemble fini  $\Sigma$  dont les éléments sont appelés *symboles*.

Un *langage*  $L$  sur  $\Sigma$  est un sous-ensemble de l'ensemble des mots  $\Sigma^*$ .

**Définition 6.1.1 :** Un *langage* est dit *rationnel*, s'il se définit à partir des opérations ensemblistes d'union, de produit cartésien et d'étoile de Kleene (1.2).



De manière équivalente, on peut déduire qu'un langage est rationnel à partir de son *monoïde syntaxique* : deux mots  $u$  et  $v$  de  $\Sigma^*$  sont *syntactiquement équivalents* pour  $L$  s'ils ont même contexte :

$$u \sim v \quad \text{ssi} \quad (x \cdot u \cdot y \in L \iff x \cdot v \cdot y \in L), \quad \text{pour tous } x, y \in \Sigma^*.$$

La relation d'équivalence  $\sim$  est une congruence (à droite et à gauche) de monoïde. Le *monoïde syntaxique* de  $L$  est le quotient  $\Sigma^* / \sim$  du monoïde libre  $\Sigma^*$  par la congruence  $\sim$ .

**Théorème 6.1.2 ([Ner58]) :** Un langage  $L$  sur l'alphabet  $\Sigma$  est *rationnel* si et seulement si son *monoïde syntaxique* est fini.

**Exemple 6.1.3 :** L'ensemble des mots qui sont l'écriture binaire d'un entier congruent à 3 modulo 7 est un langage rationnel.

**Exemple 6.1.4 :** L'ensemble des mots qui contiennent autant de  $a$  que de  $b$  n'est pas un langage rationnel.

Ce théorème caractérise un langage rationnel de manière nécessaire et suffisante comme l'ensemble des langages ayant un nombre fini de résiduels. Les résiduels sont l'ensemble des classes d'équivalences formées par la relation  $\sim$  considérée uniquement comme une congruence à droite :

$$u \sim v \quad \text{ssi} \quad (u \cdot y \in L \iff v \cdot y \in L), \quad \text{pour tout } y \in \Sigma^*.$$

Ces classes d'équivalences permettent de définir naturellement une structure d'automate dit minimal.

## 6.2. Automates

**Définition 6.2.1 :** Un *automate fini déterministe*  $\Omega$  sur l'ensemble (fini) d'états  $U$  et l'alphabet  $\Sigma$  est un triplet  $(i, A, \delta)$  où

- $i \in U$  est l'*état initial*,
- $A \subset U$  est l'ensemble des *états terminaux*,
- $\delta : U \times \Sigma \rightarrow U$  est la *fonction de transition*.

On note  $\delta^* : U \times \Sigma^* \rightarrow U$  la *fonction de transition étendue* de l'automate  $\Omega$  définie par :

$$\delta^*(q, w \cdot a) = \delta(\delta^*(q, w), a) \quad \text{et} \quad \delta^*(q, \epsilon) = q,$$

pour tout mot  $w \in \Sigma^*$  et tout symbole  $a \in \Sigma$ . On note  $\delta_q$  la fonction des transitions à partir de l'état  $q$  :

$$\delta_q : \begin{cases} \Sigma & \longrightarrow U, \\ a & \longmapsto \delta(q, a). \end{cases}$$

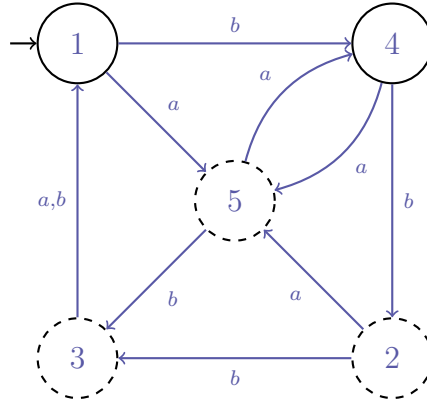
On dit que deux états  $p$  et  $q$  ont *même statut* (de terminaison) si :

$$(p \in A) = (q \in A) ,$$

c'est-à-dire si  $p$  et  $q$  sont soit tous deux terminaux soit tous deux non-terminaux.

On appelle *automate non-initial* une structure d'automate sans état initial, c'est-à-dire uniquement un couple  $(A, \delta)$  d'états terminaux et une fonction de transition.

**Exemple 6.2.2 :** La structure de graphe dirigé suivante :



encode l'automate  $(1, \{2, 3, 5\}, \delta)$  sur l'ensemble d'états  $[5]$  et l'alphabet  $\{a, b\}$  où  $\delta$  est définie par les arêtes décorées par  $a$  ou  $b$  du graphe ( $\delta(1, a) = \delta(4, a) = \delta(2, a) = 5, \delta(1, b) = \delta(5, a) = 4$ , etc.).

Le *langage reconnu* par un automate  $\Omega = (i, A, \delta)$ , noté  $L(\Omega)$ , est l'ensemble des mots pour lesquels il existe une séquence de transitions depuis l'état initial  $i$  vers un état terminal :

$$L(\Omega) = \{ w \in \Sigma^* \mid \delta^*(i, w) \in A \} .$$

**Exemple 6.2.3 :** L'automate précédent reconnaît les mots :

$$a, aaa, aaaaa, \dots, aaaab, \dots, aab, ab, ba, baaa, \dots, baab, \dots, bb, bba, \dots .$$

Le *langage droit* d'un état  $q$  de l'automate  $\Omega$  est l'ensemble des mots reconnus à partir de l'état  $q$  :

$$L_q(\Omega) = \{ w \in \Sigma^* \mid \delta^*(q, w) \in A \} .$$

**Remarque 6.2.4 :**  $L(\Omega) = L_i(\Omega)$ .

On dit que deux états  $p$  et  $q$  sont *distingués* si leur langage droit est distinct ( $L_p(\Omega) \neq L_q(\Omega)$ ). Deux états  $p$  et  $q$  sont *quasi-distingués* s'ils ont un statut de terminaison distinct ou si leurs fonctions de transition sont distinctes (c'est-à-dire si  $(p \in A) \neq (q \in A)$  ou  $\delta_p \neq \delta_q$ ).

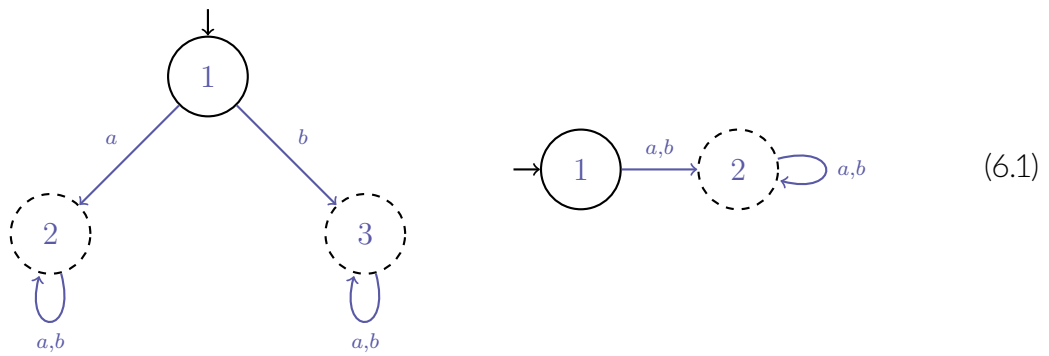
**Définition 6.2.5 :** Un automate  $\Omega$  est (*quasi-*)*distingué* si tous ces états sont (*quasi-*)*distingués*.

**Remarque 6.2.6 :** Un automate *distingué* est nécessairement *quasi-distingué*.

L'ensemble des langages reconnus par les automates forment les *langages reconnaissables*. En 1951, Stephen C. KLEENE prouve le théorème suivant :

**Théorème 6.2.7 ([Kle51]) :** L'ensemble des *langages reconnaissables* est exactement l'ensemble des *langages rationnels*.

Deux automates peuvent reconnaître le même langage. Par exemple les deux automates :



reconnaissent, tous les deux, l'ensemble des mots non-vides sur l'alphabet  $\{a, b\}$ . La notion d'*automate* (déterministe) *minimal* permet de donner une représentation canonique (à isomorphisme près pour l'action de réétiquetage des états) des langages rationnels en terme d'automates.

## 6.3. Automates minimaux

**Définition 6.3.1 :** Un automate (fini déterministe) est *minimal* s'il n'existe pas d'automate ayant strictement moins états qui reconnaît le même langage.

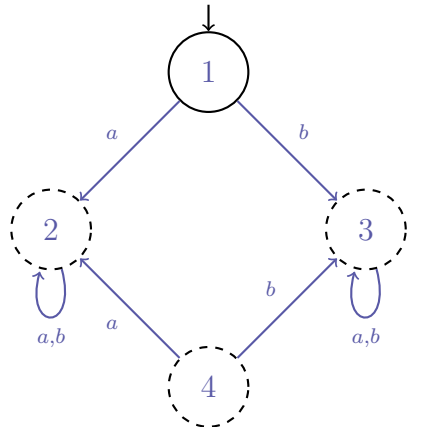
Il existe de nombreux algorithmes de minimisation permettant de calculer l'automate minimal à partir d'un automate quelconque. On peut notamment citer l'élégant algorithme de Janusz A. BRZOWSKI [Brz62], l'algorithme de Edward F. MOORE [Moo56] ou encore le plus efficace (dans le pire des cas) donné par John E. HOPCROFT [Hop71] ( $O(kn \log n)$  où  $k$  est le nombre de symboles de  $\Sigma$  et  $n$  le nombre d'états).

Les deux derniers algorithmes se basent sur l'idée de partitionner l'ensemble des états des automates à partir de la relation « avoir même langage droit ». Ce partitionnement coïncide avec la notion de résiduel du langage reconnu par l'automate et permet ainsi décrire plus finement la notion d'automate minimal.

Un état  $q$  est *accessible* s'il existe un mot  $w \in \Sigma^*$  tel que  $\delta^*(i, w) = q$ .

**Définition 6.3.2 :** Un automate est *accessible* si tous ses états sont accessibles.

**Exemple 6.3.3 :** Les précédents automates sont *accessibles* mais le suivant ne l'est pas :



L'état 4 n'est pas accessible depuis 1.

**Proposition 6.3.4 (Critère de minimalité) :** Un automate est *minimal* si et seulement s'il est à la fois *distingué* et *accessible*.

Ce résultat est une propriété de théorie des langages. On présente ici uniquement l'idée de la preuve :

PREUVE : Si l'automate  $\Omega$  n'est pas distingué ou n'est pas accessible alors on peut aisément construire un automate ayant moins d'états. (L'idée est fusionner les états reconnaissant le même langage droit.) D'où  $\Omega$  n'est pas minimal.

Réciproquement, on montre que si l'automate est distingué et accessible alors l'ensemble des (langages droits des) états est en bijection avec l'ensemble des résiduels du langage. À partir du théorème de Nérode (Théorème 6.1.2), on déduit que l'automate est donc minimal. ■

Nous utiliserons ce critère dans la suite pour déduire de nouvelles formules d'énumération des automates acycliques.

**Remarque 6.3.5 :** L'ensemble des résiduels coïncide avec l'ensemble des langages droits des états d'un automate. Le critère de minimalité permet d'une part de déduire que tous les langages droits de l'automate sont utiles (accessibilité) et d'autre part qu'ils ne sont pas redondants (distinguableté).

On appelle *partie accessible* de  $\Omega$  à partir de  $q$ , notée  $\Omega^{(q)}$ , l'automate défini avec  $q$  comme état initial et restreint à l'ensemble des états accessibles à partir de  $q$ . Formellement, la partie accessible  $\Omega^{(q)}$  est l'automate  $(q, A', \delta')$  sur l'ensemble d'états  $U'$  où

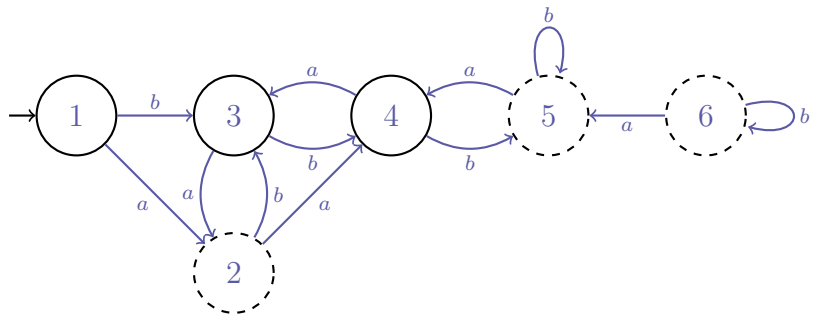
$$U' = \{ p \in U \text{ où il existe } w \in \Sigma^*, \text{ tel que } \delta^*(q, w) = p \},$$

$$A' = A \cap U',$$

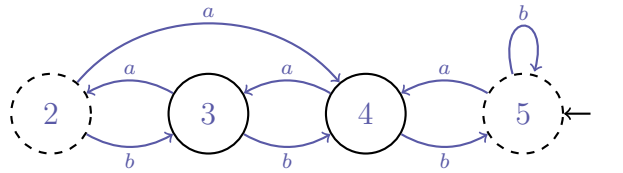
et

$$\delta' : \begin{cases} U' \times \Sigma & \longrightarrow U', \\ (p, a) & \longmapsto \delta(p, a). \end{cases}$$

**Exemple 6.3.6 :** Soit l'automate  $\Omega$  :



La partie accessible  $\Omega^{(5)}$  est l'automate  $(5, \{2, 5\}, \delta')$  :



## 6.4. Automates acycliques et langages finis

Parmi les langages rationnels, la famille des langages finis est une famille remarquable associée aux *automates acycliques*. Cette sous-famille va nous intéresser dans toute la suite de ce chapitre.

**Définition 6.4.1 :** Un automate  $(i, A, \delta)$  est *acyclique* si :

- il existe un unique état  $\theta \in U$  pour lequel  $\delta(\theta, a) = \theta$ , pour tout  $a \in \Sigma$ , et,  $\theta \notin A$ ,
- pour tout état  $q \in U \setminus \{\theta\}$  et tout mot non-vide  $w \in \Sigma^+$ , on a  $\delta^*(q, w) \neq q$ .

L'état  $\theta$  est appelé *état absorbant*.

**Remarque 6.4.2 :** Certains articles (comme [Dom04]) définissent les automates acycliques comme les automates reconnaissant un langage fini. Ce cadre ne nous convient pas, car il n'interdit pas d'avoir plusieurs états absorbants ou plusieurs composantes fortement connexes sans état terminaux.

Pour faciliter la suite, on redéfinit/adapte la précédente définition des automates pour le cas des automates acycliques en traitant l'état absorbant comme un état en dehors à notre ensemble d'états  $U$  :

**Définition 6.4.3 :** Un **automate acyclique**  $\Omega$  sur l'ensemble (fini) d'états  $U$  et l'alphabet  $\Sigma$  est un triplet  $(i, A, \delta)$  où

- $i$  est l'état initial,
- $A$  est un ensemble d'états terminaux,
- $\delta : U \times \Sigma \rightarrow U \cup \{\theta\}$  telle que

$$\delta^*(q, w) \neq q,$$

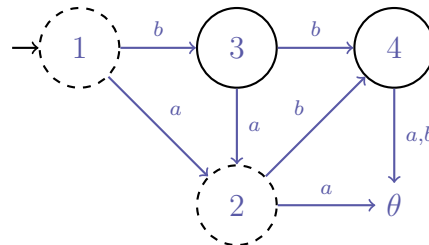
pour tout  $q \in U$  et tout mot non-vidé  $w \in \Sigma^+$ .

(La fonction de transition étendue est aussi complétée par  $\delta^*(\theta, w) = \theta$  quelque soit le mot  $w$ .) Cette définition est bien entendue équivalente à la précédente, mais le fait de sortir l'état absorbant des états va nous faciliter les choses par la suite.

On dit qu'un état  $q$  est **co-accessible** s'il existe un mot  $w$  tel que  $\delta^*(q, w) \in A$ .

**Définition 6.4.4 :** Un automate est **co-accessible** si tous ses états (excepté l'état absorbant) sont co-accessibles.

**Exemple 6.4.5 :** Les automates précédents possèdent tous des cycles et le suivant est acyclique :



Il reconnaît le langage  $\{\epsilon, a, ba\}$ . (L'état 4 est le seul état de l'automate qui n'est pas co-accessible.)

**Remarque 6.4.6 :** L'ensemble des états non co-accessibles a pour langage droit l'ensemble vide.

## 6.5. Automates acycliques minimaux

Un **automate acyclique minimal** est un automate minimal. Dans la précédente sous-section, on a donné un critère de minimalité : un automate est minimal si et seulement s'il est *distingué* et *accessible*. Dans le cas des automates acycliques, on peut facilement déduire si un automate est distingué :

**Proposition 6.5.1 :** *Un automate (fini déterministe) acyclique est distingué s'il est co-accessible et quasi-distingué.*

PREUVE: Soit  $\Omega$  un automate acyclique. D'après la Remarque 6.4.6, s'il existe  $q$  non co-accessible alors  $L_q(\Omega) = L_\theta(\Omega) = \emptyset$  et ainsi  $\Omega$  n'est pas distingué.

On pose  $h : U \rightarrow \mathbb{N}$  la fonction de hauteur qui associe, à tout état  $q$ , la longueur de ses plus grandes séquences de transitions de l'état  $q$  vers l'état absorbant :

$$h(q) := \begin{cases} -1 & \text{si } q = \theta, \\ 1 + \max \{ h(\delta(q, a)) \} & \text{pour } a \in \Sigma \end{cases} \quad \text{sinon ;} \quad (6.2)$$

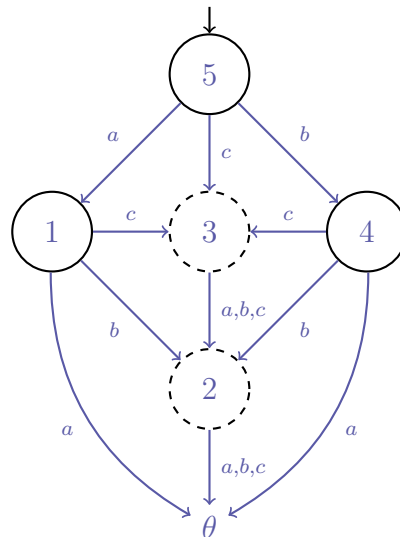
On suppose l'automate co-accessible. S'il existe plusieurs états  $p$  et  $q$  tels que  $h(p) = h(q) = 0$  alors nécessairement  $p, q \in A$  (sinon cela contredit l'hypothèse de co-accessibilité) et on a  $L_p(\Omega) = L_q(\Omega) = \{\epsilon\}$ . Dans ce cas, on déduit que l'automate n'est pas distingué. Par induction sur la longueur de ces plus grandes séquences on déduit que s'il existe deux états  $p$  et  $q$  tels que  $(p \in A) = (q \in A)$  et  $\delta_p = \delta_q$  alors l'automate n'est pas distingué. En effet, le langage droit d'un état  $p$  se déduit par l'induction sur les langages droits des états à « hauteur » plus petite :

$$L_p(\Omega) = A_p \cup \bigcup_{a \in \Sigma} \{a\} \cdot L_{\delta(p, a)}(\Omega), \quad (\text{où } h(\delta(p, a)) = h(p) - 1)$$

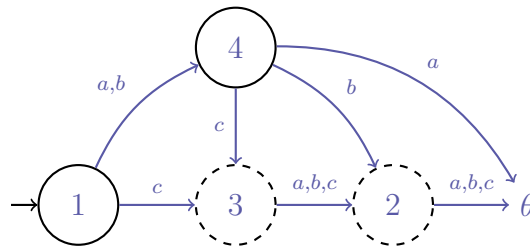
avec  $A_p = \{\epsilon\}$  si  $q \in A$  et  $\emptyset$  sinon. D'où si  $(p \in A) = (q \in A)$  et  $\delta_p = \delta_q$ , on a par induction  $L_p(\Omega) = L_q(\Omega)$ . On a ainsi montré la contraposée de notre propriété. ■

Cette proposition va être très utile dans la suite pour déduire des formules d'énumération des automates acycliques (non-initiaux) distingués.

**Exemple 6.5.2 :** L'automate :



n'est pas distingué car les états 1 et 4 sont tous deux non-terminaux et ont les mêmes fonctions de transition. Le suivant reconnaît exactement le même langage et il est distingué (il s'agit même de l'automate minimal) :





## 7. Énumération d'automates

Dans cette section, on s'intéresse à l'énumération des automates de manière générale. La prochaine section s'intéressera plus précisément au cas des automates acycliques. On note  $\mathcal{D}_k$  la donnée qui associe, à tout ensemble fini  $U$ , l'ensemble des automates finis déterministes sur l'ensemble d'états  $U$  sur un alphabet à  $k$  symboles  $\Sigma$  (fixé), noté  $\mathcal{D}_k[U]$ .

**Remarque 7.0.3 :** La donnée  $\mathcal{D}_k$  peut s'étendre en une espèce :

$$\mathcal{D}_k[\sigma]((i, A, \delta)) := (\sigma(i), \sigma(A), \delta'),$$

où

$$\delta'(q, a) := \sigma(\delta(\sigma^{-1}(q), a)),$$

pour toute bijection  $\sigma : U \rightarrow V$ .

L'énumération des automates de manière générale se révèle un problème *a priori* simple (dans le cas étiqueté). En effet, un automate sur un alphabet à  $k$  symboles n'est rien de plus qu'un triplet  $(i, A, \delta)$  où  $i$  est un élément de l'ensemble fini  $U$ ,  $A$  un sous-ensemble de  $U$ , et enfin  $\delta$  est un  $k$ -uplet d'endofonctions de l'ensemble  $U$ . Autrement dit, le nombre d'automates à  $n$  états sur un alphabet à  $k$  symboles est :

$$\#\mathcal{D}_k[n] = n 2^n n^{kn}.$$

Cela étant l'énumération à isomorphisme près fait partie des nombreux problèmes difficiles présentés par Frank HARARY dans les années soixante [Har60] et résolu avec Edgar M. PALMER [HP67]. Ce même problème a aussi été étudié par Valery A. Liskovets dans [Lis71] et Aleksej D. KORSHUNOV dans [Kor86].

De plus, le cas de l'énumération des automates minimaux demeure un problème encore ouvert [Rob85]. Il a été partiellement résolu par Frédérique BASSINO, Julien DAVID et Andrea SPORTIELLO qui ont récemment donné une asymptotique [BDS12].

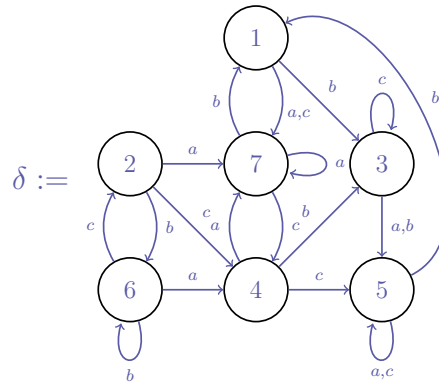
Un premier raffinement possible de ce résultat consiste à énumérer uniquement les *automates accessibles*.

### 7.1. Énumération d'automates accessibles

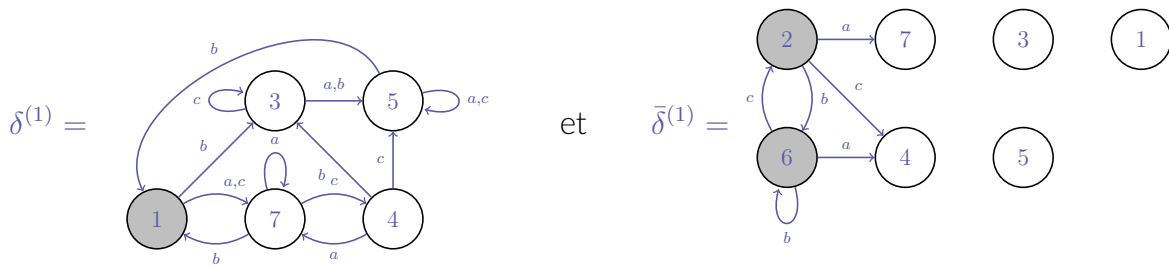
Indépendamment, Valery A. LISKOVETS [Lis69] et Robert W. ROBINSON [Rob85] utilisent une idée classique en théorie des graphes pour énumérer les automates accessibles.

Soit  $\delta : U \times \Sigma \rightarrow U$  une fonction de transition d'un automate sur l'ensemble d'états  $U$  et l'alphabet  $\Sigma$ ; et soit  $q$  un état de  $U$ . On peut découper, de manière unique, cette fonction de transition en : sa « partie accessible » à partir de  $q : \delta^{(q)}$  (c'est-à-dire la restriction de  $\delta$  aux états accessibles à partir de  $q$ ); et une application qui encode les fonctions de transition des états restants (non-accessibles à partir de  $q$ ).

**Exemple 7.1.1:** Soit l'état 1 et la fonction de transition  $\delta$  décrite par la structure de graphe :



Cette fonction  $\delta$  se découpe de manière unique comme la fonction de transition  $\delta^{(1)}$  définie comme la restriction de  $\delta$  aux états accessibles à partir de 1 et sa fonction complémentaire  $\bar{\delta}^{(1)} : \{2, 6\} \times \{a, b, c\} \rightarrow [7]$  :



Pour  $u \in U$  fixé, le découpage d'un automate quelconque sur l'ensemble d'états  $U$  est donc unique. D'un côté, on obtient l'ensemble des automates accessibles (initialement connectés en  $u$ ) sur un sous-ensemble d'états  $S \subset U$  (où  $u \in S$ ), et de l'autre, on obtient l'ensemble des fonctions de  $U \setminus S \rightarrow U$ . L'unicité du découpage décrit ainsi une bijection entre les structures :

$$\text{End}^{\times k}[U] \simeq \sum_{\substack{S \sqcup T = U \\ u \in S}} \mathcal{I}^{(u)}[S] \times \mathcal{F}[T, U],$$

où  $\text{End}^{\times k}[U]$  est l'ensemble des fonctions de transition  $\delta : U \times \Sigma \rightarrow U$  (c'est-à-dire l'ensemble des  $k$ -uplets d'endofonctions de  $U$ ),  $\mathcal{I}^{(u)}[S]$  est l'ensemble des fonctions de transition  $\delta^{(u)} : S \times \Sigma \rightarrow S$  accessibles à partir de  $u$ , et enfin,  $\mathcal{F}[T, U]$  est l'ensemble des applications  $f : T \times \Sigma \rightarrow U$ , pour tout  $u \in U$ . On sait énumérer le membre de gauche

$k$	$\mathcal{I}_n^{(u)} / (n-1)!, n = 1, 2, 3, \dots$	OEIS
2	1, 12, 216, 5248, 160675, 5931540, 256182290, 12665445248, ...	<a href="#">A006689</a>
3	1, 56, 7965, 2128064, 914929500, 576689214816, 500750172337212, ...	<a href="#">A006690</a>
4	1, 240, 243000, 642959360, 3508208993750, 34253071111894176, ...	
5	1, 992, 6903873, 175483321344, 11826519415721875, ...	

TABLE 7.1. – Le nombre d’automates accessibles à  $n$  états sur un alphabet à  $k$  symboles (divisé par la factorielle  $n - 1$ ).

(le nombre de fonctions de transition sur un ensemble de  $n$  états et un alphabet à  $k$  symboles est  $n^{kn}$ ); on sait aussi énumérer l’ensemble des applications  $f$  ( $n^{k(n-j)}$ ). Par double dénombrement, on déduit donc de cette bijection le nombre de fonctions de transitions accessibles à partir d’un état  $u$  fixé :

$$n^{kn} = \sum_{j=1}^n \binom{n-1}{j-1} \mathcal{I}_j^{(u)} n^{k(n-j)},$$

où  $\mathcal{I}_j^{(u)} = \#\mathcal{I}^{(u)}[j]$ . En multipliant par  $2^j$ , on obtient alors le nombre d’automates accessibles (initialement connecté par l’état  $u$ ).

**Remarque 7.1.2 :** Les  $-1$  dans le binomial correspondent au fait que  $u$  n’intervient pas dans le réétiquetage.

**Remarque 7.1.3 :** Dans la Table 7.1, la division par  $(n - 1)!$  permet de retrouver les séquences [\[A006689\]](#) et [\[A006690\]](#) dans l’encyclopédie des nombres [\[Slo15\]](#). Cela ne signifie pas pour autant que ces automates n’ont que des automorphismes triviaux (d’automates). En effet, si on prend comme définition d’égalité entre deux automates  $\Omega$  et  $\Lambda$  sur l’ensemble d’états  $U$  (ayant même état initial) :

$$\Omega = \Lambda \quad \text{si et seulement si} \quad L_q(\Omega) = L_q(\Lambda), \quad \text{pour tout état } q \in U,$$

alors l’automate  $\Omega$ , à gauche dans l’Équation 6.1, est fixé par la permutation 132. Cette condition d’égalité est plus forte que l’égalité entre graphes dirigés généralement utilisée dans de nombreux articles comme [\[Lis69 ; Rob85 ; DKS02 ; AMR05\]](#) (à savoir  $\Omega = \Lambda$  ssi  $\delta_q = \delta'_q$  pour tout  $q \in U$  où  $\delta$  et  $\delta'$  sont respectivement les fonctions de transition de  $\Omega$  et  $\Lambda$ ).

**Remarque 7.1.4 :** Si on considère la définition précédente d’égalité entre automates, alors le nombre d’automates minimaux (à isomorphisme près) est le coefficient de la fonction complète  $h_{1^n}$  dans la série indicatrice de cycles des automates accessibles (cf. § 1.5.2).

$k$	$\mathcal{Q}_n/n!, n = 0, 1, 2, \dots$	OEIS
1	1, 2, 6, 20, 70, 252, 924, 3432, 12870, 48620, 184756, 705432, ...	<a href="#">A000984</a>
2	1, 2, 28, 816, 35960, 2118760, 156238908, 13834413152, ...	
3	1, 2, 120, 24804, 10668000, 7817031300, 8718181624152, 13756291035998280, ...	
4	1, 2, 496, 695520, 2829877120, 25228433594000, 418757336658971856, ...	

TABLE 7.2. – Le nombre d’automates quasi-distingués (non-initiaux) à  $n$  états sur un alphabet à  $k$  symboles (divisé par la factorielle  $n$ ).

## 7.2. Énumération d’automates quasi-distingués

Un second raffinement de l’énumération des automates, tout aussi simple, est l’énumération des *automates quasi-distingués*. Cette partie bien que simple semble être nouvelle et ne pas exister dans la littérature.

On rappelle qu’un *automate quasi-distingué* est un automate pour lequel tout couple d’états  $(p, q)$ , soit  $p$  et  $q$  n’ont pas le même statut de terminaison ( $(p \in A) \neq (q \in A)$ ) soit leurs transitions sont distinctes ( $\delta_p \neq \delta_q$ ). On note  $\mathcal{Q}[U]$  l’ensemble des automates quasi-distingués sur l’ensemble d’états  $U$ .

**Remarque 7.2.1 :** Un automate acyclique qui est *quasi-distingué* est un automate *distingué* (Proposition 6.5.1).

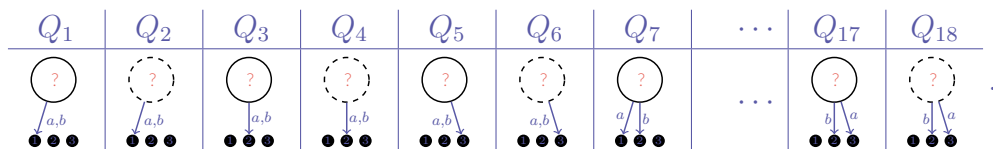
Soit  $\delta : U \times \Sigma \rightarrow U$  une fonction de transition. Pour un état  $q$  donné, ses transitions sont définies par l’application  $\delta_q : \Sigma \rightarrow U$ . Cette application est une des  $n^k$  applications possibles. Par ailleurs, l’état  $q$  est soit terminal soit non-terminal, ce qui nous amène à dire qu’un automate à  $n$  états est *quasi-distingué* si les choix des  $n$  couples  $(b_q, \delta_q)$  (où  $b_q$  est le booléen  $(q \in A)$ ) sont tous distincts parmi les  $2n^k$  choix possibles. Donc le nombre d’automates (non-initiaux) quasi-distingués à  $n$  états est :

$$\mathcal{Q}_n = n! \binom{2n^k}{n}.$$

En multipliant par  $n$  (le choix de l’état initial), on a donc le nombre d’automates quasi-distingués à  $n$  états (cf. Tables 7.2 et 7.3).

De manière équivalente, choisir l’ensemble des  $n$  couples  $(b_q, \delta_q)$  correspond à choisir  $n$  ensemble parmi  $2n^k$ . Autrement dit, les automates (non-initiaux) quasi-distingués sont en bijection l’ensemble des séquences  $(Q_i)$  de  $2n^k$  ensembles où  $\#Q_i \leq 1$ , pour tout  $i$ .

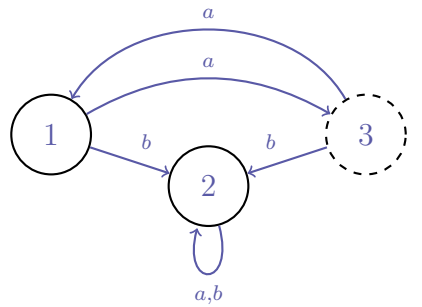
**Exemple 7.2.2 :** Pour  $U := [3]$  et  $\Sigma = \{a, b\}$ , on associe à chaque  $Q_i$  (d’une séquence de  $\mathbb{E}^{2 \cdot 3^2}$ ) une unique application de  $\Sigma \rightarrow U$  (en se fixant un ordre total quelconque sur les applications) :



$k$	$n\mathcal{Q}_n/n!$ , $n = 1, 2, 3, \dots$	OEIS
1	2, 12, 60, 280, 1260, 5544, 24024, 102960, 437580, 1847560, 7759752, ...	<a href="#">A005430</a>
2	2, 56, 2448, 143840, 10593800, 937433448, 96840892064, ...	
3	2, 240, 74412, 42672000, 39085156500, 52309089744912, ...	
4	2, 992, 2086560, 11319508480, 126142167970000, 2512544019953831136, ...	

TABLE 7.3. – Le nombre d’automates quasi-distingués à  $n$  états sur un alphabet à  $k$  symboles (divisé par la factorielle  $n$ ).

La séquence  $(\cdot \mid \cdot \mid 2 \mid \cdot \mid \cdot \mid \cdot \mid \cdot \mid \cdot \mid 3 \mid \cdot \mid \cdot \mid \cdot \mid \cdot \mid \cdot \mid \cdot \mid \cdot \mid \cdot \mid 1 \mid \cdot)$  est alors en bijection avec l’automate (non-initial) quasi-distingué :



Dans la suite sur l’énumération des automates acycliques minimaux, on utilisera la même idée sur des séquences d’ensembles bien choisies afin d’extraire une formule d’énumération des *automates acycliques distingués*.

### 7.2.1. Expérimentation : série indicatrice de cycles des automates (non-initiaux) distingués

À partir de la définition d’égalité de deux automates définie précédemment (Remarque 7.1.3), on peut calculer brutalement (ou plutôt demander à la machine de le faire pour nous) les premiers termes de la série indicatrice de cycles de l’espèce  $\mathcal{Q}$  des automates (non-initiaux) quasi-distingués. En utilisant la Définition 1.5.9 et en générant explicitement l’ensemble des points fixes de chaque permutation, on obtient les premiers caractères :

$$\begin{aligned}
\text{ch}(\mathcal{Q}, 0) &= 1, \\
\text{ch}(\mathcal{Q}, 1) &= 2p_1 = 2h_1 = 2e_1 = 2s_1, \\
\text{ch}(\mathcal{Q}, 2) &= 28p_{11} + 12p_2 = 16h_{11} + 24h_2 = 40e_{11} - 24e_2 = 16s_{11} + 40s_2 \\
\text{ch}(\mathcal{Q}, 3) &= 816p_{111} + 648p_{21} + 336p_3 = 504h_{111} + 288h_{21} + 1008h_3, \\
&= 1800e_{111} - 2304e_{21} + 1008e_3 = 504s_{111} + 1296s_{21} + 1800s_3, \\
\text{ch}(\mathcal{Q}, 4) &= 35960p_{1111} + 36304p_{211} + 11496p_{22} + 34624p_{31} + 21840p_4, \\
&= 23936h_{1111} + 10112h_{211} + 2304h_{22} + 16512h_{31} + 87360h_4,
\end{aligned}$$

$$\begin{aligned}
&= 140224e_{1111} - 309824e_{211} + 89664e_{22} + 191232e_{31} - 87360e_4, \\
&= 23936s_{1111} + 81920s_{211} + 60288s_{22} + 110848s_{31} + 140224s_4.
\end{aligned}$$

Ces caractères permettent d'extraire les premiers termes du nombre d'automates (non-initiaux) distingués (à isomorphisme près) :

$$1, 2, 16, 504, 23936, \dots$$

### 7.3. Énumération d'automates quasi-distingués accessibles

Dans les sections précédentes, on a présenté une première technique pour énumérer les automates accessibles (§ 7.1), et une seconde pour énumérer les automates quasi-distingués § 7.2. À partir des deux précédentes techniques, on peut déduire une formule d'énumération des *automates quasi-distingués accessibles*. Soit  $\Omega = (A, \delta)$  un automate quasi-distingué (non-initial) sur l'ensemble d'états  $U$ . On pose

$$J_\Omega := \{ (b_u, \delta_u) \mid b_u = (u \in A) \text{ pour tout } u \in U \}.$$

Comme l'automate est distingué, cet ensemble  $J_\Omega$  est nécessaire et suffisant pour décrire totalement  $\Omega$ . Soit  $u$  un des états. La partie accessible  $\Omega^{(u)}$  se décrit également de la même manière comme l'ensemble :

$$J_\Omega^{(u)} := \{ (b_p, \delta_p) \in J_\Omega \mid \text{où il existe } w \in \Sigma^* \text{ tel que } \delta^*(u, w) = p \}.$$

**Remarque 7.3.1 :** La partie accessible  $\Omega^{(u)}$  est un automate quasi-distingué accessible.

Le complémentaire de la partie accessible  $\Omega^{(u)}$  est un couple  $(A', \bar{\delta}^{(u)})$  qui correspond en terme d'ensemble à la différence des deux précédents :

$$\bar{J}_\Omega^{(u)} = J_\Omega \setminus J_\Omega^{(u)}.$$

Comme dans la section précédente (§ 7.1), une fois  $u$  fixé, le découpage de  $J_\Omega$  en un ensemble  $J_\Omega^{(u)}$  et un ensemble  $\bar{J}_\Omega^{(u)}$  est unique. Par double dénombrement, on peut donc déduire la formule suivante :

$$\mathcal{Q}_n = \sum_{j=1}^n \binom{n-1}{j-1} \mathcal{Q}_j^{(u)} (n-j)! \binom{2n^k - j}{n-j},$$

où

- $\mathcal{Q}_j^{(u)}$  correspond au nombre d'automates quasi-distingués accessibles,
- $(n-j)! \binom{2n^k - j}{n-j}$  correspond au nombre de manière de choisir un ensemble  $\bar{J}_\Omega^{(u)}$  (pour  $J_\Omega^{(u)}$  fixé).

$k$	$\mathcal{Q}_n^{(u)}/(n-1)!, n = 1, 2, 3, \dots$	OEIS
1	2, 6, 16, 40, 96, 224, 512, 1152, 2560, 5632, 12288, ...	<a href="#">A057711</a>
2	2, 42, 1504, 72964, 4461448, 328954456, 28379014048, ...	
3	2, 210, 60736, 32759500, 28341003672, 35883182538392, ...	
4	2, 930, 1912000, 10181374300, 111468391930168, 2180890303500256504, ...	

TABLE 7.4. – Le nombre d’automates quasi-distingués accessibles (à partir de l’état initial  $u$ ) à  $n$  états sur un alphabet à  $k$  symboles (divisé par la factorielle  $n - 1$ ).

Le binomial,  $\binom{2n^k-j}{n-j}$ , correspond au nombre de choix de  $n - j$  couples  $(b, \nu)$  parmi l’ensemble des couples restants  $2n^k - j$  (les  $j$  couples utilisés dans  $J_\Omega^{(u)}$  étant interdits, afin que  $J_\Omega^{(u)}$  et  $\bar{J}_\Omega^{(u)}$  soient disjoints). Le factoriel,  $(n - j)!$ , devant correspond au nombre de manières d’affecter ces couples aux  $n - j$  états de  $\bar{J}_\Omega^{(u)}$ . À partir de cette nouvelle formule, on obtient une meilleure borne supérieure du nombre d’automates minimaux (cf. Table 7.4) que celle proposée par Michael DOMARATZKI, Derek KISMAN et Jeffrey O. SHALLIT dans [DKS02].

Cette nouvelle formule permet d’énumérer la famille des automates quasi-distingués. Cette famille peut aisément être générée et pourrait permettre d’apporter un nouvel angle d’attaque pour répondre au problème encore ouvert de l’énumération des automates minimaux [BN07 ; BDS12].

## 8. Énumération d'automates acycliques minimaux

Le problème d'énumération des automates minimaux a longtemps été un problème ouvert. On peut tout d'abord citer Michael DOMARATZKI (avec Derek KISMAN et Jeffrey O. SHALLIT puis seul) qui s'est intéressé à fournir des bornes inférieures et supérieures du nombre d'automates acycliques minimaux [DKS02 ; Dom03 ; Dom04]. Ces bornes ont été améliorées par Cezar CÂMPEANU et Wing H. HO [CH04]. Marco ALMEIDA, Nelma MOREIRA et Rogério REIS se sont intéressés à donner une représentation en termes de mots aux automates ainsi qu'à la génération (exhaustive et aléatoire) de ces structures [AMR05 ; AMR07 ; AMR08]. De manière surprenante, David CALLAN établit une bijection entre les automates acycliques non-étiquetés (sans états terminaux) et des chemins dans certains treillis marqués ; il donne une formule d'énumération de ces automates utilisant un calcul de déterminant sur des nombres de cycles de Stirling [Cal08].

Ce problème a aussi intéressé Valery A. LISKOVETS [Lis06]. L'auteur réutilise les techniques utilisées (et citées dans le chapitre précédent) sur les automates (sans contrainte d'acyclicité) [Lis69 ; Lis71] pour compter d'une part le nombre de fonctions de transition des automates acycliques, et d'autre part le nombre d'automates acycliques accessibles.

Dans cette section, on reprend les travaux de Valery A. LISKOVETS [Lis06] sur l'énumération exacte d'automates acycliques (Définition 6.4.3). L'auteur donne une formule d'énumération récursive des automates acycliques (non-initiaux) :

$$a(k; n) = \sum_{s=1}^n (-1)^{s-1} \binom{n}{s} 2^s (n-s+1)^{ks} a(k; n-s), \quad [\text{Lis06, Théorème 3.1}]$$

avec  $a(k, 0) = 1$ . Notre travail part de la remarque que cette récurrence est un cas particulier d'une formule d'énumération des fonctions de parking généralisées fournie par Joseph P. S. KUNG et Catherine H. YAN :

$$p(\chi; n) = \sum_{s=1}^n (-1)^{s-1} \binom{n}{s} \chi(n-s-1)^s p(\chi; n-s), \quad [\text{KY03, Théorème 4.2}]$$

avec  $p(\chi; 0) = 1$ . Pour  $\chi$  définit par :

$$\chi : m \longmapsto 2m^k,$$

on retombe en effet sur la première formule. On en déduit donc :



**Théorème 8.0.2 ([Pri15a, Théorème 2]) :** *Il existe une bijection entre les automates acycliques non-initiaux sur l'ensemble (fini) d'états  $U$  et un alphabet à  $k$  symboles et les  $2m^k$ -fonctions de parking sur l'ensemble  $U$ .*

Dans cette section, on explicite une bijection entre ces structures. Cette bijection permet de traduire de nombreuses propriétés des automates directement sur les fonctions de parking. On va notamment pouvoir extraire de nouvelles formules d'énumération des automates acycliques *co-accessibles* (Table 8.1), *quasi-distingués* (et donc *distingués* (Proposition 6.5.1)) (Tables 8.2 et 8.3) et enfin des automates acycliques *minimaux* (Table 8.4).

On commence par construire une bijection entre les automates (finis déterministes) acycliques non-initiaux sur un alphabet à  $k$  symboles et les  $2m^k$ -fonctions de parking. On extrait alors, de cette bijection, des propriétés sur les langages droits des automates directement sur les fonctions de parking. On déduira alors de manière bijective des formules d'énumération des automates acycliques (non-initiaux) co-accessibles et distingués.

La bijection se généralise à une famille d'*automates acycliques étendus* possédant des *états absorbants supplémentaires*. On définit sur ces structures généralisées les notions de co-accessibilité et de quasi-distinguabilité. En utilisant les mêmes techniques que précédemment sur les automates étendus, on obtient également des formules d'énumération sur ces structures étendues co-accessibles et quasi-distinguées. On étend une dernière fois la bijection afin d'ajouter suffisamment de contraintes. Cette dernière extension permet de donner une formule de double énumération analogue à celles des articles [Lis69 ; Lis06] (respectivement utilisées pour l'énumération des automates et automates acycliques accessibles). On déduit alors de cette formule le nombre d'*automates acycliques accessibles et distingués*, c'est-à-dire le nombre d'*automates acycliques minimaux*.

## 8.1. Bijection entre les automates acycliques (non-initiaux) et les fonctions de parking généralisées

D'après les formules d'énumération [Lis06, Théorème 3.1] et [KY03, Théorème 4.2], il existe une bijection entre les automates acycliques (non-initiaux) sur un ensemble fini d'états  $U$  et un alphabet à  $k$  symboles et les  $2m^k$ -fonctions de parking sur l'ensemble  $U$  (Théorème 8.0.2).

La bijection qui suit peut se voir comme une généralisation de la bijection entre les fonctions de parking (usuelles) et les forêts d'arbres enracinés de Dominique FOATA et John RIORDAN [FR74] (cf. § 5.3 avec l'espèce des ensembles  $\mathbf{E}$  et la courbe discrète identité). Évidemment, dans le cas où  $k = 1$ , les automates acycliques (non-initiaux) coïncident avec les forêts d'arbres enracinés (et bicolorés sur les arêtes). Autrement dit, les automates acycliques (non-initiaux) sur un alphabet à un symbole sont en bijection

avec les structures de l'espèce :

$$\mathcal{F}^{(2,2)} = \mathbb{E} \left( X \cdot \mathcal{F}^{(2,2)} \right)^2 .$$

Ce résultat découle immédiatement des Théorèmes 5.3.18 et 8.0.2.

**Remarque 8.1.1 :** La bijection qui suit utilise implicitement la notion de hauteur introduite dans la preuve de la Proposition 6.5.1. Dans le cas où l'alphabet contient un unique symbole, cette notion correspond à la hauteur dans l'arbre. Dans le cas des arbres, cette hauteur peut s'interpréter, dans l'équation fonctionnelle que satisfait la série

$$\mathcal{F}^{(2,2)}(t) = \exp \left( t \mathcal{F}^{(2,2)}(t) \right)^2 ,$$

comme une « hauteur » de composition de la fonction exponentielle. Intuitivement, chaque étiquette à un niveau  $h$  est rattachée par une arête à une autre étiquette au niveau  $h - 1$ . Dans le cas des automates acycliques sur un alphabet à  $k \geq 2$  symboles, on aimerait traduire en terme de « composition » de fonction le fait de rattacher une étiquette au niveau  $h$  à  $k$  étiquettes réparties sur différents niveaux entre 0 et  $h - 1$ . Une telle traduction permettrait d'obtenir une équation fonctionnelle des séries génératrices des automates acycliques.

### 8.1.1. Ordre de parking et ordre alphabétique

Soit  $(Q_i)$  une  $2m^k$ -fonction de parking sur l'ensemble  $U$  à  $n$  éléments. On se fixe une fois pour toutes  $\ell \ll U$  un ordre total sur l'ensemble d'étiquettes et on suppose un ordre total sur l'alphabet  $\Sigma$  :

$$a_1 <_{\Sigma} a_2 <_{\Sigma} \cdots <_{\Sigma} a_k .$$

La première étape de la bijection consiste à reprendre la construction de l'ordre total  $<_q$  sur les fonctions de parking (5.10) :

$$u_1 <_q u_2 <_q \cdots <_q u_n ,$$

défini par

$$u <_q v \iff \begin{cases} u <_{\ell} v & \text{et } u, v \in U_i, \\ u \in U_i, v \in U_j & \text{et } i < j. \end{cases}$$

Cet ordre permet par la suite de construire une fonction de transition satisfaisant la contrainte d'acyclicité  $\delta^*(q, w) \neq q$  quelque soit le mot  $w \in \Sigma^+$  (et l'état  $q \neq \theta$  avec  $q \in U$ ). On augmente cet ordre en y ajoutant l'état absorbant  $\theta (\notin U)$  :

$$\theta =: u_0 <_q u_1 <_q u_2 <_q \cdots <_q u_n .$$

**Exemple 8.1.2:** Soit  $(Q_i)$  la  $2m^k$ -fonction de parking sur  $[6]$  :

[illegible]

où les  $Q_i$  d'indice  $i$  impair sont sur la première ligne et ceux d'indice pair sont sur la seconde ( $Q_2 = \{5\}$ ,  $Q_5 = \{4\}$ ,  $Q_8 = \{3\}$ , etc.). L'ordre  $<_q$  associé est

$$\theta <_q 5 <_q 4 <_q 3 <_q 6 <_q 2 <_q 1.$$

### 8.1.2. Factorisation de parking et fonctions de transition

La seconde étape factorise la fonction de parking en  $n$  facteurs  $F_j$  consécutifs :

$$F_j := (Q_i) \quad \text{avec} \quad 2^j k < i \leq 2(j+1)^k, \quad \text{pour} \quad 0 \leq j < n.$$

Dans la suite, on va considérer différentes familles de fonctions de parking. On sera donc amené à modifier cette définition de la factorisation. Plus généralement, la factorisation d'une  $\chi$ -fonction de parking  $(Q_i)$  sur l'ensemble  $U$  à  $n$  éléments correspond aux  $n$  facteurs  $(F_i)$  :

$$F_j := (Q_i) \quad \text{avec} \quad \chi(j) < i \leq \chi(j+1), \quad \text{pour} \quad 0 \leq j < n. \quad (8.1)$$

Cette généralisation de la factorisation va permettre dans la suite d'utiliser le même algorithme de construction des automates indépendamment de la famille de parking considérée.

**Exemple 8.1.3 :** La précédente fonction de parking ( $Q_i$ ) de l'Exemple 8.1.2 se découpe en six facteurs :

[illegible]

On construit des ensembles de couple  $(b, \nu)$ , statut de terminaison–fonction de transition, en bijection avec ces facteurs  $F_j$ . À partir de l’ordre fixé sur l’alphabet  $\Sigma$ , on peut identifier les applications  $\nu : \Sigma \rightarrow U$  à des mots  $\nu(a_1)\nu(a_2) \cdots \nu(a_k)$ . On commence par poser l’ensemble des mots qu’on identifiera aux applications  $\nu$  et associera à un facteur  $F_j$  :

$$W_j := \{ p_1 p_2 \cdots p_k \in \{u_0, u_1, \dots, u_j\}^k \text{ où il existe } i \text{ tel que } p_i = u_j \} . \quad (8.2)$$

Cet ensemble  $W_j$  sera utilisé pour construire l'ensemble des transitions de chaque état présent dans le facteur  $F_j$  d'une  $2m^k$ -fonction de parking.

**Proposition 8.1.4 :** À l'étiquette  $\theta$  près, pour tout  $0 \leq j < n$ , les étiquettes/lettres de l'ensemble  $\{u_0, u_1, \dots, u_j\}$  des mots de  $W_j$  apparaissent dans des ensembles  $Q$  présents dans les facteurs  $F_i$  où  $i < j$ .

PREUVE : Par définition, l'ordre  $<_q$  se définit en lisant chaque  $Q_i$  de  $i = 1$  à  $2n^k$  (de gauche à droite). On déduit alors immédiatement de la contrainte de parking :

$$\sum_{j=1}^{2i^k} \#Q_j \geq i, \quad \text{pour tout } 1 \leq i \leq n,$$

que l'ensemble d'étiquettes est inclus dans les ensembles qui précèdent le facteur  $F_j$  :

$$\{u_0, u_1, \dots, u_j\} \subset \{\theta\} \cup \bigcup_{1 \leq i \leq 2(j+1)^k} Q_i. \quad \blacksquare$$

Cette proposition nous assure, dans la prochaine étape, que la construction de la fonction de transition est acyclique.

Chaque facteur  $F_i$  est par définition de longueur  $2((i+1)^k - i^k)$ , soit deux fois plus que le cardinal de l'ensemble  $W_i$ . L'objectif est d'associer à chaque ensemble de  $F_i$  un mot (qui correspondra à une transition) et un statut d'acceptation. On pose alors

$$W_i^+ := \{(b, w) \mid b \text{ un booléen et } w \in W_i\}.$$

Ce dernier ensemble de mots signés va alors permettre de définir l'ensemble des couples  $(b_q, \delta_q)$  dans l'automate (où  $b_q$  est le statut de terminaison de  $q$  et  $\delta_q$  sa fonction de transition).

**Proposition 8.1.5 :**

$$\#W_i = (i+1)^k - i^k \quad \text{et} \quad \#W_i^+ = 2\#W_i.$$

PREUVE : Le nombre de mots de longueur  $k$  sur un alphabet  $A$  à  $i+1$  symboles est  $(i+1)^k (= \#A^k)$ . Les mots de  $W_i$  sont les mots sur un alphabet à  $i$  symboles contenant au moins une fois un certain symbole fixé  $u$ . Ceci est équivalent à la différence des mots de  $A^k$  avec les mots ne contenant pas  $u$  :  $(A \setminus \{u\})^k$ . D'où

$$\#W_i = \#A^k \setminus (A \setminus \{u\})^k = \#A^k - \#(A \setminus \{u\})^k = (i+1)^k - i^k.$$

Par définition, l'ensemble  $W_i^+$  est  $W_i$  dédoublé où à chaque mot de  $W_i$  est associé vrai et faux. ■

Pour tout  $0 \leq i < n$ , on a maintenant autant de mots signés  $(b, w) \in W_i^+$  que d'ensembles dans le facteur  $F_i$ . Pour un ordre fixé (quelconque) sur l'ensemble des mots, on peut donc appareiller chaque mot  $w$  à un unique ensemble  $Q_j$  du facteur. À partir de l'ordre  $<_q$  définit précédemment, on a automatiquement un ordre total lexicographique, aussi noté  $<_q$ , sur l'ensemble  $W_i$ . On construit l'ordre suivant sur  $W_i^+$  :

$$(b, w) <_+ (b', w') \quad \text{si et seulement si} \quad (w = w' \text{ et } b = \text{faux}) \quad \text{ou} \quad (w \neq w' \text{ et } w <_q w'),$$

pour tous  $(b, w)$  et  $(b', w')$  distincts.

Cet ordre permet de mettre en bijection les ensembles d'un facteur  $F_i$  avec les mots signés de  $W_i^+$ .

**Remarque 8.1.6 :** Le choix de l'ordre  $<_+$  sur  $W_i^+$  n'est pas significatif pour la suite. Cela étant nous attirons l'attention du lecteur sur le fait qu'il est utilisé dans toute la suite afin de faciliter les différentes constructions. En particulier, on utilisera implicitement cet ordre pour extraire l'ensemble associé au couple  $(faux, \nu)$  où  $\nu : a \mapsto \theta$  définit toutes les transitions vers l'état absorbant.

**Exemple 8.1.7 :** Les ensembles  $Q_i$  de chaque facteur  $F_j$  de l'exemple précédent sont associés à un unique mot signé  $(b, w)$  de  $W_j^+$  où  $b$  est associé soit à la première ligne si  $b = vrai$  soit à la seconde si  $b = faux$  en correspondance à l'ordre  $<_+$ , et  $w$  est représenté en colonne sous les (deux) ensembles associés :

$$\begin{aligned}
F_0 &= \left( \begin{array}{c} \cdot \\ 5 \end{array} \right), & F_1 &= \left( \begin{array}{c|c|c} \cdot & 4 & \cdot \\ \cdot & \cdot & 3 \end{array} \right), & F_2 &= \left( \begin{array}{c|c|c|c|c} \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \end{array} \right), \\
W_0^+ &= \left\{ \begin{array}{c} \theta \\ \theta \end{array} \right\}, & W_1^+ &= \left\{ \begin{array}{c|c|c} \theta & 5 & 5 \\ 5 & \theta & 5 \end{array} \right\}, & W_2^+ &= \left\{ \begin{array}{c|c|c|c|c} \theta & 5 & 4 & 4 & 4 \\ 4 & 4 & \theta & 5 & 4 \end{array} \right\}, \\
F_3 &= \left( \begin{array}{c|c|c|c|c|c|c} \cdot & \cdot & \cdot & \cdot & 2 & \cdot & \cdot \\ \cdot & \cdot & 6 & \cdot & \cdot & \cdot & \cdot \end{array} \right), & F_4 &= \left( \begin{array}{c|c|c|c|c|c|c|c|c|c} \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{array} \right), \\
W_3^+ &= \left\{ \begin{array}{c|c|c|c|c|c|c} \theta & 5 & 4 & 3 & 3 & 3 & 3 \\ 3 & 3 & 3 & \theta & 5 & 4 & 3 \end{array} \right\}, & W_4^+ &= \left\{ \begin{array}{c|c|c|c|c|c|c|c|c|c} \theta & 5 & 4 & 3 & 6 & 6 & 6 & 6 & 6 & 6 \\ 6 & 6 & 6 & 6 & \theta & 5 & 4 & 3 & 6 & 6 \end{array} \right\}, \\
F_5 &= \left( \begin{array}{c|c|c|c|c|c|c|c|c|c|c|c} \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 \end{array} \right), \\
W_5^+ &= \left\{ \begin{array}{c|c|c|c|c|c|c|c|c|c|c|c} \theta & 5 & 4 & 3 & 6 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 & 2 & \theta & 5 & 4 & 3 & 6 & 2 & 2 \end{array} \right\}.
\end{aligned}$$

Dans la sous-section précédente, on a supposé fixé l'ordre  $<_\Sigma$  sur l'alphabet  $\Sigma$ . À partir de cet ordre, on représente sans ambiguïté chaque application de  $\nu : \Sigma \rightarrow U$  comme un mot :

$$\nu \simeq \nu(a_1)\nu(a_2)\cdots\nu(a_k).$$

**Proposition 8.1.8 :** L'ensemble des fonctions  $\nu : \Sigma \rightarrow \{u_0, u_1, \dots, u_i\}$  telles que  $u_i \in \nu(\Sigma)$  est en bijection avec  $W_i$ .

L'ordre lexicographique  $<_q$  sur les séquences d'états et l'ordre sur les symboles de l'alphabet  $<_\Sigma$  permettent ainsi d'appareiller de manière unique un couple  $(b, \nu)$  de booléen et d'application à chaque ensemble d'états  $Q_i$  de chaque fonction de parking.

Nous avons maintenant tous les ingrédients pour construire un automate acyclique (non-initial) sur un alphabet à  $k$  symboles  $\Sigma$ , à partir d'une  $2m^k$ -fonction de parking.

On pose alors  $\zeta_{(Q_i)}$  l'application qui associe, à chaque indice  $i$  de chaque ensemble  $Q_i$ , un unique couple  $(b, \nu)$ .

$$\begin{aligned}\zeta_{(Q_i)}(2) &= (\text{vrai}, \nu), & \text{avec } \nu &\simeq \theta\theta, \\ \zeta_{(Q_i)}(5) &= (\text{faux}, \nu), & \text{avec } \nu &\simeq 5\theta, \\ \zeta_{(Q_i)}(8) &= (\text{vrai}, \nu), & \text{avec } \nu &\simeq 55.\end{aligned}$$
[illegible]
$$\begin{aligned}\zeta_{(Q'_i)}(1) &= (\text{vrai}, \theta \theta), \\ \zeta_{(Q'_i)}(4) &= (\text{vrai}, \theta 1).\end{aligned}$$
$$A = \{ q \in U \mid q \in Q_j \text{ et } \zeta_{(Q_i)}(j) = (\text{vrai}, \nu) \} ,$$
$$\delta_q = \nu, \quad \text{avec} \quad (b, \nu) = \zeta_{(Q_i)}(j) \quad \text{pour} \quad q \in Q_j.$$
$$A = \bigcup_{\substack{1 \leq i \leq 2n^k \\ j \text{ pair}}} Q_j,$$
$$\delta_q = \nu_{2(j^k+i)} := p_1 p_2 \cdots p_k, \quad \text{avec} \quad \begin{cases} p_1 p_2 \cdots p_k & \text{le } i^{\text{ième}} \text{ mot de } W_j, \\ q \in Q_{2(j^k+i)} \cup Q_{2(j^k+i)+1} & . \end{cases}$$

132

---

**Algorithme 8.1.1 :** Fonction de parking vers automates acycliques (non-initial)

---

**Données :** une  $2m^k$ -fonction de parking sur l'ensemble  $U$  à  $n$  éléments (ordonné) et  $\Sigma$  un alphabet de  $k$  symboles

**Résultat :**  $(A, \delta)$  un automate acyclique (non-initial) sur l'ensemble d'états  $U$  et sur l'alphabet  $\Sigma$

```
1  $A \leftarrow \emptyset$ ;  
2 définir l'ordre  $<_q$  en parcourant  $(Q_i)$ ;  
3 pour chaque  $0 \leq j < n$  faire  
4   construire et ordonner  $W_j^+$ ;  
5   pour chaque  $1 \leq i \leq 2(j+1)^k - 2j^k$  faire  
6      $(b, \nu) \leftarrow$  le  $i^{\text{ième}}$  élément de  $W_j^+$ ;  
7     pour tous les  $q \in Q_{2j^k+i}$  faire  
8       si  $b$  alors ajouter  $q$  à  $A$ ;  
9        $\delta_q \leftarrow \nu$ ;  
10  $\delta \leftarrow \bigcup_{p \in U} \delta_p$ ;  
11 retourner  $(A, \delta)$ 
```

---

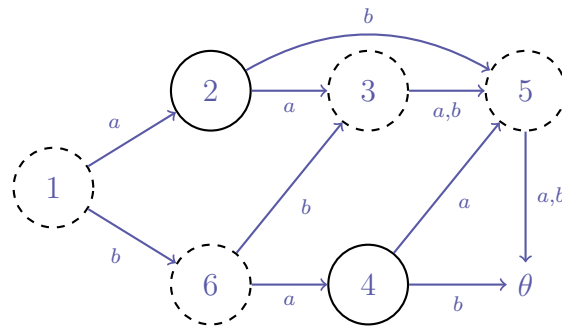
**Remarque 8.1.10 :** On s'intéresse dans la suite à des déformations de la famille des  $2m^k$ -fonctions de parking, comme les  $2m^k - 1$ -fonctions de parking. L'algorithme s'adapte aussi à ces  $\chi$ -fonctions de parking en modifiant les conditions des Lignes 5 et 7 respectivement par  $1 \leq i \leq \chi(j+1) - \chi(j)$  et  $q \in Q_{\chi(j)+i}$ . (On redéfinira explicitement  $W_j$  en fonction des nouveaux ordres  $<_q$ .)

**Remarque 8.1.11 :** Cet algorithme a une complexité en temps de l'ordre de  $\Theta(2n^k)$ . Si l'on représente les fonctions de parking sous forme de mot (cf. § 3.3), on peut calculer l'automate de manière plus efficace. En prenant l'inverse de la standardisation (§ 12.1), on calcule l'ordre  $<_q$ , puis la valeur  $w_m$ , en chaque position. Ceci correspond à l'indice de l'ensemble contenant  $m$ . On peut aisément décomposer cet entier sous la forme  $2j^k + i$  et calculer en temps linéaire le  $i^{\text{ième}}$  couple statut de terminaison-mot/fonction de transition de  $W_j^+$ .

**Théorème 8.1.12 :** Cet algorithme est une bijection les  $2m^k$ -fonctions de parking et les automates acycliques (non-initiaux) sur un alphabet à  $k$  symboles.

**PREUVE :** Par construction, le couple obtenu  $(A, \delta)$  est un automate. La Proposition 8.1.4 nous assure que la construction de  $\delta$  vérifie la contrainte d'acyclicité. L'automate est défini facteur par facteur. L'indice du facteur correspond à la hauteur des états construits dans l'automate (cf. Proposition 6.5.1). Par induction, facteur par facteur, on déduit alors que cet algorithme est injectif. Le Théorème 8.0.2 d'énumération nous assure donc qu'il s'agit d'une bijection. ■

**Exemple 8.1.13 :** L'automate obtenu sur l'alphabet  $\{a, b\}$  et l'ensemble d'états  $[5]$  à partir de la  $2m^2$ -fonction de parking des Exemples précédents 8.1.2, 8.1.3 et 8.1.7 est :



Nous ne définissons pas la construction réciproque, car elle n'a pas d'intérêt pour le propos. Cela étant, à partir de la fonction de hauteur discutée dans la preuve de la Proposition 6.5.1, on peut retrouver l'ordre  $<_q$  par induction. Chaque état à hauteur  $j$  va être associé à un unique ensemble  $Q_i$  défini dans le facteur  $j$  à partir du statut de terminaison et de sa fonction de transition.



### Exemple 8.1.14 :

		$F_0 :$
1	•	$\theta \theta \theta$
2	2	
		$F_1 :$
3	•	$\theta \theta 2$
4	4	
5	•	$\theta 2 \theta$
6	•	
7	•	$\theta 2 2$
8	•	
9	•	$2 \theta \theta$
10	•	
11	•	$2 \theta 2$
12	•	
13	•	$2 2 \theta$
14	•	
15	•	$2 2 2$
16	•	
		$F_2 :$
17	•	$\theta \theta 4$
18	•	
19	•	$\theta 2 4$
20	•	
21	•	$\theta 4 \theta$
22	•	
23	•	$\theta 4 2$
24	•	
25	•	$\theta 4 4$
26	•	
27	•	$2 \theta 4$
28	•	
29	•	$2 2 4$
30	•	
31	35	$2 4 \theta$
32	•	
33	•	$2 4 2$
34	•	
35	•	$2 4 4$
36	•	
		$\vdots$
53	•	$4 4 4$
54	•	
		$F_3 :$
55	•	$\theta \theta 3$
56	•	
		$\vdots$
		$F_4 :$
129	•	$\theta \theta 5$
130	•	
		$\vdots$
151	1	$2 4 5$
152	•	
		$\vdots$
171	6	$4 3 5$
172	•	
		$\vdots$
		$F_5 :$
		$\vdots$
		$F_6 :$
		$\vdots$
547	•	$5 1 6$
548	7	
		$\vdots$
609	8	$6 2 3$
610	•	
		$\vdots$

À gauche, une  $2m^3$ -fonction de parking ( $Q_i$ ) sur l'ensemble  $[8]$  (représentée à la verticale pour une question d'espace). On détaille chaque étape de l'algorithme de construction de l'automate sur l'alphabet  $\{a, b, c\}$  en bijection avec cette fonction de parking. À droite de la fonction de parking, on a associé les transitions des états de chaque ensemble  $Q_i$ . Les états présents dans des ensembles d'indices impairs sont non-terminaux et ceux d'indices pairs sont terminaux.

### Ordres

On considère l'ordre naturel sur  $[8]$  et celui-ci  $a <_\Sigma b <_\Sigma c$  sur l'alphabet. L'ordre  $<_q$  obtenu par lecture de gauche à droite (ici de haut en bas) de ( $Q_i$ ) est :

$$\theta <_q 2 <_q 4 <_q 3 <_q 5 <_q 1 <_q 6 <_q 7 <_q 8.$$

### Factorisation et fonction de transition

On associe à  $F_0$  l'unique fonction de transition  $\nu : \Sigma \rightarrow \{\theta\}$ . Le premier ensemble est mis en bijection avec (*vrai*,  $\nu$ ) et le second avec (*faux*,  $\nu$ ).

On a associé à  $F_1$  l'ensemble des fonctions  $\nu : \Sigma \rightarrow \{\theta, 2\}$  telles que  $2 \in \nu(\Sigma)$ . À partir de l'ordre  $<_q$ , on construit et ordonne l'ensemble  $W_1$  sur l'alphabet  $\{\theta, 2\}$  :

$$\theta\theta 2, \theta 2\theta, \theta 22, 2\theta\theta, 2\theta 2, 222,$$

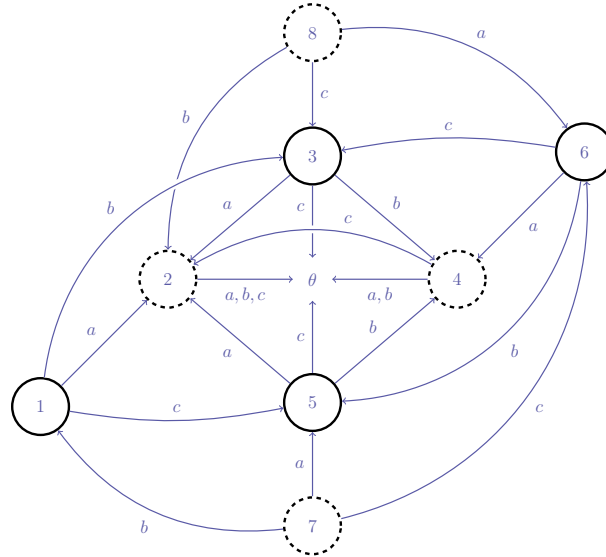
À partir de l'ordre  $<_\Sigma$ , on associe à tout mot  $p_1 p_2 p_3$  de  $W_i$  une fonction de transition  $\nu$  unique définie par :

$$\nu : a_i \mapsto p_i, \quad \text{pour tout } 1 \leq i \leq 3.$$

Chaque ensemble d'états est associé à chaque couple  $(b, \nu)$ . L'état 4 présent dans l'ensemble  $Q_4$  est associé au couple (*vrai*,  $\theta\theta 2$ ). Et ainsi de suite, jusqu'à  $F_7$ .

### Automate acyclique (non-initial)

L'algorithme construit l'automate suivant :



Dans cette sous-section, on a construit une bijection entre les automates acycliques (non-initiaux) et une famille de fonctions parking généralisées. Cette dernière nous permet maintenant d'utiliser toute la panoplie de formules d'énumération sur ces fonctions définie dans la Partie I précédente. En particuliers, l'adaptabilité du paramètre  $\chi$  de ces fonctions de parking généralisées va permettre d'énumérer la sous-famille des automates acycliques (non-initiaux) *co-accessibles* ainsi que les différentes familles d'*automates acycliques étendus* (§ 8.4). À partir de l'interprétation du caractère de ces fonctions dans la base des fonctions complètes (§ 2.5), on obtient une formule d'énumération des automates acycliques distingués. Enfin, en appliquant la même technique que précédemment (§ 7.3), on va pouvoir déduire une formule d'énumération des automates acycliques accessibles et distingués, c'est-à-dire énumérer les *automates acycliques minimaux*.

## 8.2. Co-accessibilité

La bijection associe à chaque état d'un même ensemble, dans la fonction de parking, le même couple  $(b, \nu)$ . Cela signifie que ces états ont même statut de terminaison et même fonction de transition dans l'automate. Dans la section d'introduction à la théorie des automates (Chapitre 6), on a vu qu'un automate acyclique est *distinct* s'il est *co-accessible* et *quasi-distingué* (Proposition 6.5.1). Dans cette section, on s'intéresse à obtenir la famille des automates acycliques (non-initiaux) co-accessibles. Pour se faire, on extrait une nouvelle famille de fonctions de parking associée à ces automates.

Un automate acyclique est *co-accessible* si l'ensemble de ces états (autre que l'état absorbant) accèdent à un état terminal :

pour tout état  $q \in U$ , il existe un mot  $w \in \Sigma^*$  tel que  $\delta^*(q, w) \in A$ .

La contrainte d'acyclicité oriente les transitions des états en direction de l'état absorbant  $\theta$ . On distingue trois différentes familles d'états ayant des transitions atteignant  $\theta$  :

- les états  $q$  tels que  $\delta_q(\Sigma) \cap U \neq \emptyset$ ,
- les états  $q$  tels que  $\delta_q(\Sigma) = \{\theta\}$  et  $q \notin A$ , et,
- les états  $q$  tels que  $\delta_q(\Sigma) = \{\theta\}$  et  $q \in A$ .

Les états de la première famille accèdent nécessairement à au moins une des deux suivantes. La seconde est, par définition, une famille d'*états non co-accessibles* et la troisième est une famille d'*états co-accessibles*.

Dans la précédente construction d'un automate à partir d'une fonction de parking, la première famille se compose des états présents dans les facteurs  $F_j$  avec  $j > 0$ . Les deux suivantes se composent des états du premier facteur. Plus précisément (*modulo* l'ordre utilisé pour la construction), la seconde famille est exactement l'ensemble des états de  $Q_1$  et la troisième famille est exactement l'ensemble des états de  $Q_2$ .

**Proposition 8.2.1 :** Soit  $(Q_i)$  une  $2m^k$ -fonction de parking.

L'automate acyclique associé à  $(Q_i)$  est **co-accessible** si et seulement si  $Q_1 = \emptyset$ .

PREUVE : Par définition de la hauteur d'un état (Proposition 6.5.1), chaque état à hauteur  $h > 0$  accède à au moins un état à hauteur  $h - 1$  (ainsi qu'à d'autres à hauteur inférieure). On déduit que chaque état à hauteur  $h > 0$  accède nécessairement par un mot non-vide à un état à hauteur 0. Par définition de la construction les états à hauteur  $h > 0$  proviennent des facteurs  $F_j$  avec  $j > 0$ . Si les états  $p$ , à hauteur 0, sont uniquement des états terminaux alors tous les états sont **co-accessibles**. Donc l'automate associé à une fonction de parking où  $Q_1 = \emptyset$  est **co-accessible**. Réciproquement par contraposée, si  $Q_1 \neq \emptyset$ , on a créé des états de la seconde catégorie ( $\delta_q(\Sigma) = \{\theta\}$  et  $q \notin A$ ) donc l'automate ne peut être pas **co-accessible**. ■

Cette proposition nous amène à considérer l'ensemble des  $2m^k$ -fonctions de parking  $(Q_i)$  telles que  $Q_1 = \emptyset$ . On souhaite alors simplement considérer des fonctions de parking pour lesquelles les facteurs  $F_0$  ne contiennent qu'un ensemble, et les facteurs suivants demeurent inchangés.

**Lemme 8.2.2 :** Les  $(2m^k - 1)$ -fonctions de parking sont en bijection avec les **automates acycliques (non-initiaux) co-accessibles**.

PREUVE : On utilise la factorisation généralisée des  $(2m^k - 1)$ -fonctions de parking (8.1) et on pose :

$$W_0^+ = \{(\text{vrai}, \nu)\} \quad \text{avec} \quad \nu : a \mapsto \theta \quad \text{pour tout} \quad a \in \Sigma.$$

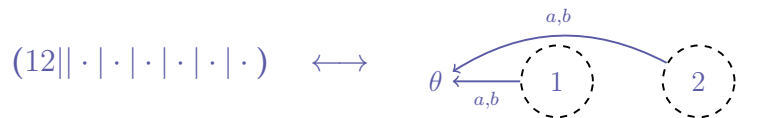
La construction d'automates, à partir des  $(2m^k - 1)$ -fonctions de parking décrites dans la Remarque 8.1.10, construit bien un automate acyclique et cette construction demeure bien injective. Par définition, les  $(2m^k - 1)$ -fonctions de parking sont en bijection avec les  $2m^k$ -fonctions de parking  $(Q_i)$  telles que  $Q_1 = \emptyset$ . La construction de l'automate a été adaptée pour « simuler » la présence d'un  $Q_1$  vide. La Proposition 8.2.1 nous assure que de plus la construction est surjective. ■

À partir de la formule de Catherine H. YAN [KY03, Théorème 4.2], on a alors la formule d'énumération récursive des **automates acycliques (non-initiaux) co-accessibles** à  $n$  états sur un alphabet à  $k$  symboles :

$$a_c(k; n) = \sum_{s=1}^n (-1)^{s-1} \binom{n}{s} (2(n-s+1)^k - 1)^s a_c(k; n-s),$$

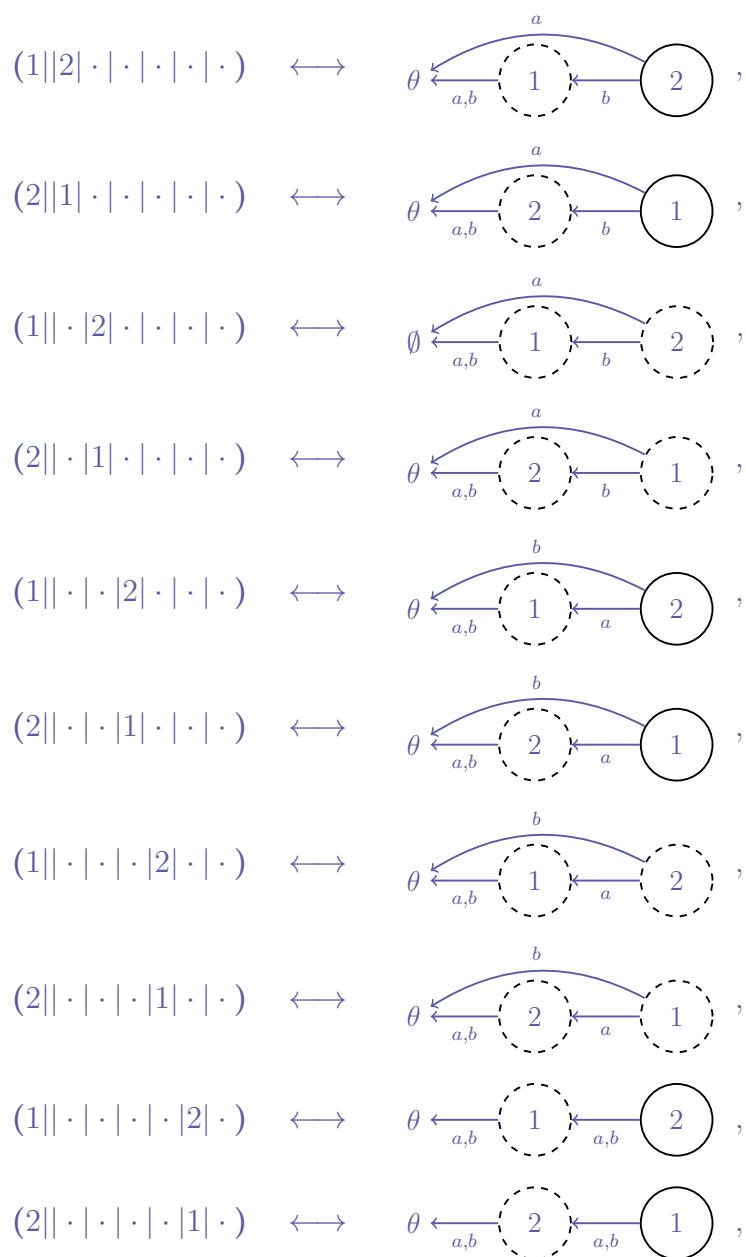
avec  $a_c(k; 0) = 1$  (cf. Table 8.1).

**Exemple 8.2.3 :** Les 13  $(2m^2 - 1)$ -fonctions de parking sur  $[2]$  et les automates acycliques (non-initiaux) co-accessibles associés :



$k$	$a_c(k; n), n = 1, 2, 3, \dots$	OEIS
1	1, 2, 12, 128, 2000, 41472, 1075648, 33554432, 1224440064, ...	<a href="#">A097629</a>
2	1, 2, 28, 1136, 95056, 13723392, 3064697536, 983685177344, ...	
3	1, 2, 60, 8192, 3177296, 2723265792, 4415947148992, ...	
4	1, 2, 124, 54128, 91590736, 435810204672, 4775080503918784, ...	

TABLE 8.1. – Le nombre d’automates acycliques (non-initiaux) co-accessibles à  $n$  états sur un alphabet à  $k$  symboles.



$$\begin{aligned}
(1|| \cdot | \cdot | \cdot | \cdot | \cdot | 2) &\longleftrightarrow \theta \xleftarrow{a,b} \textcircled{1} \xleftarrow{a,b} \textcircled{2} , \\
(2|| \cdot | \cdot | \cdot | \cdot | \cdot | 1) &\longleftrightarrow \theta \xleftarrow{a,b} \textcircled{2} \xleftarrow{a,b} \textcircled{1} .
\end{aligned}$$

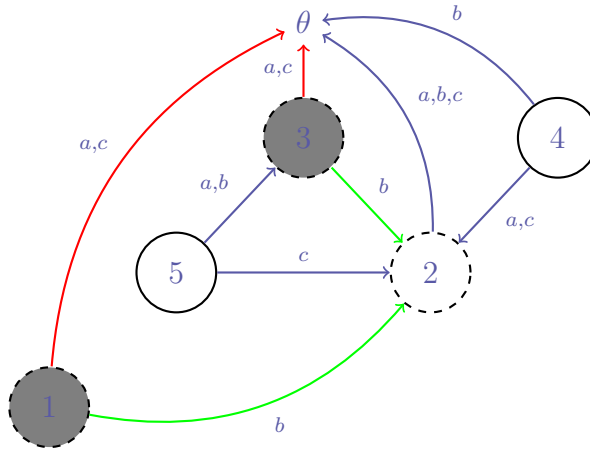
### 8.3. Quasi-distinguabilité et distinguabilité

Après avoir énuméré les automates acycliques (non-initiaux) co-accessibles, on souhaite énumérer les *automates acycliques (non-initiaux) quasi-distingués*, et, par prolongement de la précédente sous-section, on souhaite aussi énumérer les *automates acycliques (non-initiaux) distingués* (Proposition 6.5.1). La construction définit un automate ayant deux états non quasi-distingués s'ils proviennent d'un même ensemble de la fonction de parking.

**Exemple 8.3.1 :** La  $(2m^3 - 1)$ -fonction de parking  $(Q_i)$  où

$$Q_1 = \{2\}, Q_4 = \{1, 3\}, Q_8 = \{4\} \text{ et } Q_{110} = \{5\},$$

définit, par la bijection, l'automate :



où les états 1 et 3 proviennent du même ensemble  $Q_4$  et sont tous deux terminaux avec les mêmes transitions.

Soit  $(Q_i)$  une  $2m^k$ -fonction de parking (resp. une  $(2m^k - 1)$ -fonction de parking) sur un ensemble à  $n$  éléments, et, soit  $\Omega$  l'automate acyclique (non-initial) associé.

**Proposition 8.3.2 :** L'automate  $\Omega$  est *quasi-distingué* si et seulement si  $\#Q_i \leq 1$ , pour tout  $1 \leq i \leq 2n^k$  (resp.  $i \leq 2n^k - 1$ ).

Comme chaque ensemble est associé à un couple  $(b, \nu)$  unique de  $W_i^+$ , cette proposition est immédiate.

On cherche donc à extraire la sous-famille des  $2m^k$  et  $(2m^k - 1)$ -fonctions de parking où les ensembles sont au plus de cardinalité 1. Dans la partie précédente sur les fonctions de parking généralisées (Partie I), on a donné une interprétation du caractère des fonctions de parking généralisées dans les fonctions symétriques (non-commutatives, cf. § 2.5, Théorème 4.2.7 et [PV15, Théorème 3.4]). Le caractère des  $\chi$ -fonctions de parking, sur un ensemble  $U$  à  $n$  éléments, est une somme sur les compositions de l'entier  $n$  :

$$\text{ch}(\mathcal{P}_\chi; n) = \sum_{\pi \models n} \alpha(\pi) \mathbf{S}^\pi,$$

où  $\alpha(\pi)$  est donné par la récurrence (Proposition 4.2.9) :

$$\alpha(\pi) = \sum_{i=1}^j (-1)^{i-1} \alpha(\pi_1 \pi_2 \cdots \pi_{j-i}) \binom{\chi(\pi(j-i) + 1) - j + i}{i},$$

pour toute composition  $\pi := \pi_1 \pi_2 \cdots \pi_j$  de  $n \geq 1$  et vaut 1 pour  $\pi$  l'unique composition de l'entier 0.

Chaque terme  $\alpha(\pi)$  compte le nombre de  $\chi$ -fonctions de parking (à isomorphisme près) sur  $U$  dont la sous-séquence des cardinaux des ensembles non-vides correspond à  $\pi$  (§ 2.5). Les  $\chi$ -fonctions de parking où  $Q_i \leq 1$  quelque soit  $i$  sont ainsi comptées par  $\alpha(1^n)$ .

### 8.3.1. Quasi-distingués

À partir de la bijection entre les automates acycliques (non-initiaux) et les  $2m^k$ -fonctions de parking, on déduit alors le nombre d'*automates acycliques (non-initiaux) quasi-distingués* par la récurrence suivante :

$$a_q(k; n) = \sum_{i=1}^n (-1)^{i-1} \binom{n}{i} a_q(k; n-i) i! \binom{2(n-i+1)^k - n + i}{i},$$

avec  $a_q(k, 0) = 1$  (Table 8.2).

PREUVE : Par spécialisation de  $\mathbf{S}^\pi$  sur le multinomial  $\binom{n}{\pi_1, \dots, \pi_j}$  et d'après la bijection, on a :

$$a_q(k; n) = \alpha(1^n) \binom{n}{1, \dots, 1} = \alpha(1^n) n!,$$

d'où, par définition de  $\alpha$  :

$$a_q(k; n) = n! \sum_{i=1}^n (-1)^{i-1} \alpha(1^{n-i}) \binom{2(n-i+1)^k - n + i}{i},$$

$k$	$a_q(k;n)/n!, n = 1, 2, 3, \dots$	OEIS
1	1, 2, 5, 14, 42, 132, 429, 1430, 4862, 16796, 58786, 208012, ...	<a href="#">A000108</a>
2	1, 2, 13, 166, 3324, 92718, 3354712, 150206430, 8050991676, ...	<a href="#">A177448</a>
3	1, 2, 29, 1298, 124706, 21256346, 5723038913, 2251528402754, ...	
4	1, 2, 61, 8830, 3727540, 3543721650, 6467723423732, ...	

TABLE 8.2. – Le nombre d’automates acycliques (non-initiaux) quasi-distingués à  $n$  états sur un alphabet à  $k$  symboles (divisé par  $n!$ ).

$k$	$a_d(k;n)/n!, n = 1, 2, 3, \dots$	OEIS
1	1, 1, 2, 5, 14, 42, 132, 429, 1430, 4862, 16796, 58786, 208012, ...	<a href="#">A000108</a>
2	1, 1, 6, 75, 1490, 41415, 1495806, 66908663, 3584202810, ...	
3	1, 1, 14, 623, 59766, 10182221, 2740931060, 1078233863119, ...	
4	1, 1, 30, 4335, 1829410, 1739056185, 3173912127246, 9981391304293375, ...	

TABLE 8.3. – Le nombre d’automates acycliques (non-initiaux) distingués à  $n$  états sur un alphabet à  $k$  symboles (divisé par  $n!$ ).

$$\begin{aligned}
&= \sum_{i=1}^n (-1)^{i-1} n! \frac{a_q(k; n-i)}{(n-i)!} \binom{2(n-i+1)^k - n + i}{i}, \\
&= \sum_{i=1}^n (-1)^{i-1} \binom{n}{i} i! a_q(k; n-i) \binom{2(n-i+1)^k - n + i}{i}. \quad \blacksquare
\end{aligned}$$

### 8.3.2. Distingués : co-accessibles et quasi-distingués

De la même manière, en utilisant la bijection entre les automates acycliques (non-initiaux) co-accessibles et les  $(2m^k - 1)$ -fonctions de parking (Lemme 8.2.2) et l’interprétation du caractère de Frobenius (Proposition 4.2.9) de ces fonctions de parking, on déduit que le nombre d’*automates acycliques (non-initiaux) distingués* (co-accessibles et quasi-distingués, Proposition 6.5.1) est compté par :

$$a_d(k; n) = \sum_{i=1}^n (-1)^{i-1} \binom{n}{i} a_d(k; n-i) i! \binom{2(n-i+1)^k - 1 - n + i}{i}, \quad (8.3)$$

avec  $a_d(k, 0) = 1$  (Table 8.3). En conclusion à ces deux dernières sous-sections, on a obtenu des formules d’énumération de différentes familles d’automates acycliques en interprétant des propriétés des automates directement sur les fonctions de parking. Après avoir caractérisé les fonctions de parking associées aux automates co-accessibles et celles associées aux automates quasi-distingués, puis aux automates dis-

tingués, il nous reste à déduire comment extraire les automates acycliques minimaux. Pour cela, on introduit les automates acycliques étendus :

## 8.4. Automates acycliques étendus

L'idée de cette section est d'introduire les outils suffisants pour utiliser la même technique que pour l'énumération des automates (sans contraintes d'acyclicité) quasi-distingués (§ 7.3). Comme la partie accessible  $\Omega^{(i)}$  d'un automate acyclique distingué  $\Omega$  est un automate minimal, on introduit les automates étendus pour caractériser la partie complémentaire de cette partie accessible  $\bar{\Omega}^{(i)}$ .

Dans [Lis06], V. A. LISKOVETS introduit les automates acycliques étendus comme des automates ayant des états absorbants supplémentaires :

**Définition 8.4.1 :** Un **automate acyclique étendu** (non-initial) sur l'ensemble d'états  $U$ , l'ensemble d'états absorbants supplémentaires  $T$  et l'alphabet  $\Sigma$ , est un couple  $(A, \delta)$  où

- $A \subset U$  est l'ensemble des états terminaux,
- $\delta : U \times \Sigma \rightarrow U \cup \{\theta\} \cup T$  tel que

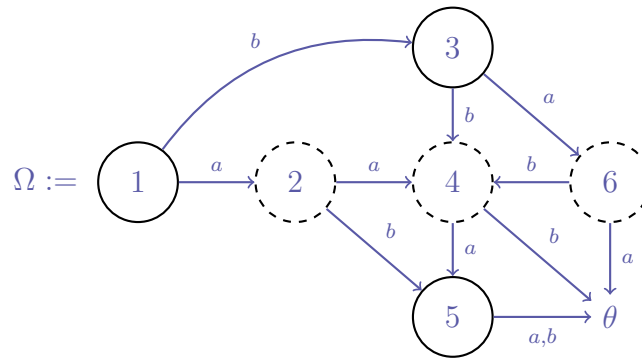
$$\delta^*(q, w) \neq q, \quad \text{pour tout } q \in U \text{ et tout mot non-vide } w \in \Sigma^+.$$

(La fonction de transition étendue  $\delta^*$  se comporte avec les états absorbants supplémentaires comme avec l'état absorbant usuel, c'est-à-dire  $\delta^*(t, w) = t$  pour tout mot  $w$  et tout état absorbant  $t \in \{\theta\} \cup T$ .)

**Remarque 8.4.2 :** Pour  $T = \emptyset$ , un automate acyclique étendu est un automate acyclique.

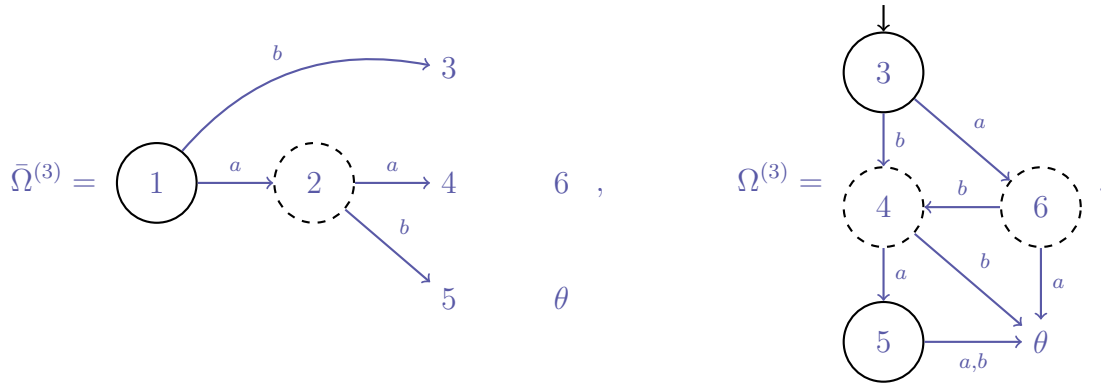
Ces automates étendus permettent de caractériser le complémentaire de la partie accessible d'un automate acyclique.

**Exemple 8.4.3 :** Soit l'automate acyclique :





La partie accessible à partir de 3 et son complémentaire sont respectivement les structures à droite et à gauche :



L'automate acyclique étendu, à gauche, est sur l'ensemble d'états  $\{1, 2\}$  et l'ensemble d'états absorbants supplémentaires  $T := \{3, 4, 5, 6\}$  et l'automate acyclique, à droite, est sur l'ensemble d'états  $T$ .

L'auteur de [Lis06] donne une formule d'énumération de ces structures généralisant à la précédente [Lis06, Théorème 3.1] en introduisant le paramètre  $t$  correspondant au nombre d'états absorbants supplémentaires :

$$e(k, t; n) = \sum_{s=1}^n (-1)^{s-1} \binom{n}{s} 2^s (n - s + t + 1)^{ks} e(k, t; n - s), \quad [\text{Lis06, Théorème 3.1}]$$

Encore une fois, cette formule est un cas particulier de [KY03, Théorème 4.2] pour  $\chi : m \mapsto 2(m + t)^k$ .

**Lemme 8.4.4 :** Les  $(2(m+t)^k)$ -fonctions de parking sont en bijection avec les automates acycliques étendus (non-initiaux) avec  $t$  états absorbants supplémentaires sur un alphabet de  $k$  symboles.

En appliquant le découpage en facteur des fonctions de parking (8.1), on obtient :

$$\begin{aligned} \ell(F_0) &= 2(t+1)^k, \\ \ell(F_1) &= 2((t+2)^k - (t+1)^k), \\ \ell(F_3) &= 2((t+3)^k - (t+2)^k), \\ &\vdots \\ \ell(F_{n-1}) &= 2((t+n)^k - (t+n-1)^k), \end{aligned}$$

pour  $n$  le nombre d'états (non-absorbants) et  $t$  le nombre d'états absorbants supplémentaires. Avec les mêmes idées que précédemment, on adapte la bijection. La valeur  $(t+1)^k$  correspond exactement au nombre d'applications  $\Sigma \rightarrow \{\theta\} \cup T$  de l'alphabet

On suppose un ordre fixé sur l'ensemble  $T$  des états absorbants supplémentaires :

On augmente l'ordre  $<_q$  sur  $U$  avec ces états supplémentaires :

puis on redéfinit les ensembles de mots  $W_j$  associés aux facteurs  $F_j$  (8.2) par :

et pour  $i \geq 1$ ,

$$W_i = \{ p_1 p_2 \cdots p_k \in \{\theta_1, \dots, \theta_t, \theta, u_1, \dots, u_i\}^k \mid \text{où il existe } i \text{ tel que } p_i = u_i \}.$$

L'Algorithme 8.1.1 (adapté avec la Remarque 8.1.10) fournit ainsi une bijection entre ces structures.

**Exemple 8.4.6 :** Soit  $(Q_i)$  la  $2(m+1)^k$ -fonction de parking sur  $[3]$  :

On pose  $\theta_1$  l'état absorbant supplémentaire. En appliquant la bijection à  $(Q_i)$ , on obtient l'automate acyclique étendu sur l'ensemble d'états [3], l'ensemble d'états absorbants supplémentaires  $\{\theta_1\}$  et l'alphabet  $\{a, b\}$  :



### 8.4.1. Automates étendus co-accessible

L'idée des prochaines définitions (*automate étendu co-accessible* et *distingué*) est d'obtenir une caractérisation des complémentaires des parties accessibles des automates co-accessibles et quasi-distingués. On souhaite, pour un automate acyclique quasi-distingué (ou minimal) donné  $\Omega$ , pouvoir énumérer l'ensemble des automates étendus possibles pouvant être obtenus comme le complémentaire de  $\Omega$ .

**Définition 8.4.7 :** Un automate étendu  $(A, \delta)$  est **co-accessible** s'il ne contient pas d'état  $q$  tel que

$$q \notin A \quad \text{et} \quad \delta_q(\Sigma) = \{\theta\}.$$

Cette définition est équivalente à : l'automate étendu est *co-accessible* s'il peut se définir comme un complémentaire (quelconque) d'une partie accessible d'un automate acyclique co-accessible.

En appliquant la même technique que précédemment (§ 8.2), c'est-à-dire en retirant un ensemble dans le premier facteur des  $2(m+t)^k$ -fonctions de parking, on supprime alors la possibilité d'avoir un état terminal dont toutes les transitions accèdent à l'état absorbant lors de la construction de l'automate. On obtient alors

**Lemme 8.4.8 :** Les  $(2(m+t)^k - 1)$ -fonctions de parking sont en bijection avec les automates étendus à  $t$  états absorbants supplémentaires sur un alphabet à  $k$  symboles.

### 8.4.2. Automates étendus quasi-distingués

On étend la définition d'*automate quasi-distingué* aux automates étendus : un automate étendu est *quasi-distingué* si ses états sont *quasi-distingués*.

**Remarque 8.4.9 :** Contrairement aux automates acycliques (non-étendus), un automate étendu *quasi-distingué* n'est pas nécessairement *distingué*.

Comme précédemment (§ 8.3.1 et § 8.3.2), on déduit les formules suivantes d'énumération des automates étendus quasi-distingués et des automates étendus co-accessibles et quasi-distingués à  $t$  états absorbants supplémentaires, respectivement notés  $e_q$  et  $e_{c,q}$  :

$$e_q(k, t; n) = \sum_{i=1}^n (-1)^{i-1} \binom{n}{i} e_q(k, t; n-i) i! \binom{2(n-i+t+1)^k - n + i}{i},$$

et

$$e_{c,q}(k, t; n) = \sum_{i=1}^n (-1)^{i-1} \binom{n}{i} e_{c,q}(k, t; n-i) i! \binom{2(n-i+t+1)^k - 1 - n + i}{i},$$

où  $e_q(k, t; 0) = e_{c,q}(k, t; 0) = 1$ . On rappelle que ces formules découlent immédiatement de la Proposition 4.2.9.

### 8.4.3. Automates étendus avec contrainte

Nous avons maintenant « presque » tous les ingrédients nécessaires pour énumérer les *automates acycliques minimaux*, en utilisant la même technique que celle développée dans la sous-section traitant de l'énumération des automates (non nécessairement acycliques) quasi-distingués et accessibles (§ 7.3).

Soit  $\Omega = (A, \delta)$  un automate acyclique minimal sur l'ensemble d'états  $T$ . On pose  $b_t := (t \in A)$  pour tout  $t \in T$ .

**Définition 8.4.10 :** Un automate étendu co-accessible et quasi-distingué  $\Lambda = (A', \delta')$ , ayant  $T$  comme ensemble d'états absorbants supplémentaire et  $U$  comme ensemble d'états, satisfait la *contrainte*  $\Omega$  si

$$b_s \neq b_u \quad \text{et} \quad \delta_s \neq \delta'_u,$$

pour tous les couples  $(s, u)$  d'états respectivement de  $\Omega$  et  $\Lambda$ .

Quelque soit l'automate minimal  $\Omega$  fixé sur l'ensemble d'états  $T$ , le nombre d'automates étendus sur l'ensemble d'états absorbants supplémentaires  $T$  satisfaisant  $\Omega$  est invariant. En effet, le nombre d'automates étendus avec contrainte correspond à l'ensemble des automates étendus où l'on a interdit  $t$  couples  $(b, \nu)$  de statut de terminaison et fonction de transition.

Ces transitions interdites sont exclusivement des transitions dirigées vers les états absorbants  $(\{\emptyset\} \cup T)$ . D'après la bijection, ces  $t$  couples correspondent aux choix de  $t$  ensembles dans le premier facteur d'une  $(2(m+t)^k - 1)$ -fonction de parking. Autrement dit, si ces  $t$  ensembles sont vides dans la fonction de parking, alors l'automate étendu  $\Lambda$  associé satisfait la contrainte  $\Omega$ .

Pour un automate minimal  $\Omega$  fixé, par le même raisonnement que pour les automates acycliques (non-initiaux) co-accessibles, on obtient que :

**Lemme 8.4.11 :** Les  $(2(m+t)^k - t - 1)$ -fonctions de parking sont en bijection avec les automates étendus satisfaisant la contrainte  $\Omega$ .

Ce lemme se démontre de la même manière que le Lemme 8.2.2 (donnant une bijection entre les  $(2m^k - 1)$ -fonctions de parking et les automates acycliques (non-initiaux) co-accessibles).

Encore une fois à partir de la Proposition 4.2.9, on déduit la formule suivante d'énumération des automates étendus satisfaisant la contrainte  $\Omega$  (sur un ensemble à  $t$  états) :

$$c(k, t; n) = \sum_{i=1}^n (-1)^{i-1} \binom{n}{i} c(k, t; n-i) i! \binom{2(n-i+t+1)^k - t - 1 - n + i}{i}. \quad (8.4)$$

**Remarque 8.4.12 :** On a l'égalité suivante :

$$a_c(k, 0; n) = a_d(k; n).$$

## 8.5. Automates acycliques minimaux

Tous les ingrédients sont maintenant en place pour compter le nombre d'*automates acycliques minimaux* par double dénombrement. Soit  $i$  un état fixé de l'ensemble fini  $U$ . Quelque soit l'automate acyclique distingué  $\Omega$  sur l'ensemble d'états  $U$ , d'après le critère de minimalité (Proposition 6.3.4), on sait que la partie accessible  $\Omega^{(i)}$  est un *automate acyclique minimal*; et la partie complémentaire  $\bar{\Omega}^{(i)}$  est un *automate étendu satisfaisant la contrainte*  $\Omega^{(i)}$  par définition. Ce découpage étant unique, on en déduit alors la bijection suivante :

$$\mathcal{A}_d[U] \simeq \sum_{\substack{S \sqcup T = U \\ i \in T}} \sum_{\Omega \in \mathcal{M}[T]} \{\Omega\} \times c^{(\Omega)}[S, T],$$

où  $\mathcal{A}_d[U]$  est l'ensemble des automates acycliques distingués sur l'ensemble d'états  $U$ ,  $\mathcal{M}[T]$  est l'ensemble d'automates acycliques minimaux sur l'ensemble d'états  $T \subset U$ , et  $c^{(\Omega)}[S, T]$  est l'ensemble des automates étendus satisfaisant la contrainte  $\Omega$  sur l'ensemble d'états  $S$  et l'ensemble d'états absorbants  $T$ .

D'une part, on sait énumérer les *automates acycliques distingués* (8.3); d'autre part, on sait aussi énumérer les *automates étendus avec contrainte* (8.4). Comme le nombre d'automates étendus avec contrainte ne dépend que du nombre d'états qui définissent l'automate utilisé comme contrainte, on déduit alors de la bijection précédente :

**Théorème 8.5.1 ([Pri15b, Théorème 4]):** *Le nombre d'automates (finis déterministes) acycliques minimaux  $m(k; n)$  à  $n$  états sur un alphabet à  $k$  symboles vérifie la formule suivante :*

$$a_d(k; n) = a_c(k, 0; n) = \sum_{t=1}^n \binom{n-1}{t-1} m(k, t) c(k, t; n-t).$$

$n \backslash k$	1	2	3	4
1	1	1	1	1
2	2	6	14	30
3	4	60	532	3900
4	8	900	42644	1460700
5	16	18480	6011320	1220162880
6	32	487560	1330452032	1943245777800
7	64	15824880	428484011200	5307146859111120
8	128	612504240	190167920278448	23025057433925970000
9	256	27619664640	111649548558856000	149780070423407303443200
10	512	1425084870240	84001095774695390816	1396395902225576206029949920
11	1024	82937356685760	78954926089415009686528	17993790111404399137868446737600

TABLE 8.4. – Le nombre d'automates acycliques minimaux à  $n$  états sur un alphabet à  $k$  symboles (divisé par  $(n - 1)!$ ).

**Troisième partie .**

**Constructions d'algèbres de Hopf  
combinatoires**

The Hitch Hiker's Guide to the Galaxy offers this definition of the word "Infinite".

*Infinite* : Bigger than the biggest thing ever and then some. Much bigger than that in fact, really amazingly immense, a totally stunning size, "wow, that's big", time. Infinity is just so big that by comparison, bigness itself looks really titchy. Gigantic multiplied by colossal multiplied by staggeringly huge is the sort of concept we're trying to get across here.

(Douglas ADAMS, *The Restaurant at the End of the Universe*)

Les *algèbres de Hopf* ont été introduites en topologie algébrique pour généraliser la notion de groupe dans les années quarante et cinquante. Elles apparaissent maintenant dans de nombreux contextes : *topologie, théorie des représentations, physique, etc..* En combinatoire, elles apportent une généralisation des séries génératrices. Par exemple,

$$\text{Bal}(t) = \frac{1}{2 - \exp(t)} = 1 + t + 3\frac{t^2}{2!} + 13\frac{t^3}{3!} + 75\frac{t^4}{4!} + 541\frac{t^5}{5!} + 4683\frac{t^6}{6!} + \dots \quad [\text{A000670}]$$

est la série génératrice des partitions ordonnées d'ensembles. Cette dernière se relève dans les fonctions symétriques (comme la série indicatrice de cycles) :

$$\begin{aligned} Z_{\text{Bal}} &= \frac{1}{2 - Z_{\text{E}}} = \frac{1}{2 - \sigma_1(\mathbb{X})} = \sum_{n \geq 0} \sum_{\lambda \vdash n} \binom{\ell(\lambda)}{\lambda^{(1)}, \dots, \lambda^{(n)}} h_{\lambda}, \\ &= 1 + h_1 + h_{11} + h_2 + h_{111} + 2h_{21} + h_3 + h_{1111} + 3h_{211} + h_{22} + 2h_{31} + h_4 + \dots \end{aligned}$$

et se relève encore dans les fonctions symétriques non-commutatives, **NCSF** (comme un caractère de Frobenius des ballots pour l'action de l'algèbre 0-Hecke) :

$$\begin{aligned} Z_{\text{Bal}} &= \frac{1}{2 - \sigma_1(\mathbb{A})} = \sum_{n \geq 0} \sum_{\pi \models n} \mathbf{S}^{\pi}, \\ &= 1 + \mathbf{S}^1 + \mathbf{S}^{11} + \mathbf{S}^2 + \mathbf{S}^{111} + \mathbf{S}^{21} + \mathbf{S}^{12} + \mathbf{S}^3 + \mathbf{S}^{1111} + \mathbf{S}^{31} + \mathbf{S}^{13} + \mathbf{S}^{22} + \dots \end{aligned}$$

faisant disparaître totalement les coefficients. Ces deux relèvements font apparaître des relèvements des fonctions spéciales, comme l'exponentielle avec  $\sigma_t(\mathbb{X})$  et  $\sigma_t(\mathbb{A})$ . Ces fonctions spéciales amènent des questions naturelles comme « comment calculer un inverse ? » et les réponses ne sont pas forcément évidentes.

Prenons l'exemple de l'exponentielle dans l'algèbre **FQSym**, l'algèbre des fonctions quasi-symétriques libres [MR95] :

$$\exp(\mathcal{A}) := \sum_{n \geq 0} \mathbf{G}_{12 \dots n},$$

où  $(\mathbf{G}_{\sigma})$  est la base fondamentale duale. Cette définition est un relèvement de l'exponentielle  $\sigma_1(\mathbb{A})$  dans **NCSF**. À partir de la réalisation polynomiale des  $\mathbf{G}_{\sigma}$ , on calcule



l'inverse de cette exponentielle. Nous verrons que l'utilisation de la réalisation polynomiale simplifie grandement les calculs.

Chaque élément  $\mathbf{G}_\sigma$  se réalise comme une série dans l'algèbre libre (étendue)  $\mathbb{k}\langle \mathcal{A} \rangle$  (où  $\mathcal{A}$  est un alphabet totalement ordonné infini) :

$$\mathbf{G}_\sigma(\mathcal{A}) = \sum_{\substack{w \in \mathcal{A}^* \\ \text{std}(w) = \sigma}} w ,$$

où  $\text{std}$  est la standardisation (§ 12.1). Dans le cas des permutations identités  $12 \cdots n$ , les réalisations correspondent aux séries des mots croissants de longueur  $n$  sur l'alphabet  $\mathcal{A}$  :

$$\mathbf{G}_{12 \cdots n}(\mathcal{A}) = \sum_{a_1 \leq a_2 \leq \cdots \leq a_n} a_1 a_2 \cdots a_n .$$

L'exponentielle correspond donc au produit des séries :

$$\exp(\mathcal{A}) = \sum_{n \geq 0} \mathbf{G}_{12 \cdots n}(\mathcal{A}) = \prod_{a \in \mathcal{A}}^{\rightarrow} \frac{1}{1 - a} .$$

On obtient immédiatement l'inverse :

$$\exp(\mathcal{A})^{-1} = \prod_{a \in \mathcal{A}}^{\leftarrow} (1 - a) ,$$

et correspond à la somme de mots strictement décroissants :

$$\exp(\mathcal{A})^{-1} = \sum_{n \geq 0} (-1)^n \sum_{a_1 > a_2 > \cdots > a_n} a_1 a_2 \cdots a_n ,$$

c'est-à-dire

$$\exp(\mathcal{A})^{-1} = \sum_{n \geq 0} (-1)^n \mathbf{G}_{n(n-1) \cdots 1}(\mathcal{A}) .$$

En quelques lignes, le problème est donc résolu simplement par l'utilisation des réalisations polynomiales. Cette technique a été très largement développée par le *phalanstère de Combinatoire* de Marne-la-Vallée.

Les réalisations polynomiales d'algèbres de Hopf s'inspirent des fonctions symétriques [Mac98] et des fonctions quasi-symétriques [Ges84 ; Gel+95]. De nombreuses algèbres de Hopf sont réalisables de cette manière : l'algèbre des fonctions quasi-symétriques libres, **FQSym** (aussi appelée algèbre de Hopf des permutations) [MR95 ; DHT02], l'algèbre de Hopf des tableaux (ou algèbre des fonctions symétriques libres) **FSym** [PR95 ; DHT02], l'algèbre de Hopf des mots symétriques, **WSym** [Wol36 ; RS06], l'algèbre de Hopf des mots quasi-symétriques, **WQSym** (ou appelé aussi algèbre des fonctions quasi-symétriques non-commutatives, **NCQSym**) [Hiv99 ; NT06 ; BZ09],

l'algèbre de Hopf des fonctions de parking **PQSym** [NT04a], l'algèbre de Hopf des arbres binaires, **PBT** [LR98 ; HNT05], etc..

Chacune de ces différentes algèbres se définit à partir d'une application particulière  $\phi_{\mathcal{A}}$ , dite *partitionnante*, des mots vers des structures combinatoires de  $\mathcal{C}$ . Par exemple, on peut définir **FQSym** à partir de la standardisation qui associe, à chaque mot, un ordre linéaire (ou une « permutation »). Ces applications permettent de construire  $\mathcal{H}_{\phi}(\mathcal{A})$ , une sous-algèbre de l'algèbre libre  $\mathbb{k}\langle\mathcal{A}\rangle$ , définie à partir des fibres de  $\phi_{\mathcal{A}}$ . Les structures de coalgèbre et de bialgèbre s'obtiennent à partir de l'astuce de dédoublement d'alphabets.

L'objectif de cette partie est de formaliser les propriétés suffisantes de ces *applications partitionnantes*. On commence par rappeler quelques bases sur les algèbres de Hopf ainsi que les catégories qui vont nous permettre de formaliser la notion de *type d'alphabet*. Dans le second chapitre, on formalise les caractéristiques de ces *applications partitionnantes*. Dans le troisième chapitre, on explicite la construction de l'algèbre de Hopf (et de sa duale) associée(s) à une application partitionnante. Le quatrième chapitre fournit quelques extensions à ce formalisme. On intègre la notion de produit  $\#$  et d'(*co*/*bi*-)algèbre (*co*-)dendriforme au formalisme. Le cinquième chapitre donne quelques applications à ce formalisme. En particulier, on s'intéresse à une généralisation de la *parkisation* [NT04a] aux  $\chi$ -fonctions de parking (§ 12.2), et on montre que cette application est une *application partitionnante* si et seulement si  $\chi(m) = 1 + a(m - 1)$ , avec  $a \in \mathbb{N}_{>0}$ .

Cette partie développe l'article [HP15] avec mon directeur Florent Hivert.

## 9. Quelques prérequis

Avant de présenter le formalisme des applications partitionnantes, ce chapitre introduit quelques rappels sur les notions d'algèbre libre, algèbre libre étendue, algèbre des polynômes, théorie des catégories et algèbre de Hopf.

### 9.1. Algèbre libre large et produit tensoriel

#### 9.1.1. Algèbres libres

Soit  $\mathcal{A}$  un alphabet (cf. § 1.1.6).

**Définition 9.1.1 :** L'**algèbre libre** sur  $\mathcal{A}$  est un triplet  $\mathbb{k}\langle\mathcal{A}\rangle := (\mathbb{k}\mathcal{A}^*, \mu, \iota)$  où

- $\mathbb{k}\mathcal{A}^*$  est le  $\mathbb{k}$ -espace vectoriel des combinaisons linéaires de mots ;
- $\mu : \mathbb{k}\mathcal{A}^* \otimes \mathbb{k}\mathcal{A}^* \rightarrow \mathbb{k}\mathcal{A}^*$  est le produit défini linéairement à partir du produit de concaténation :

$$\mu.(u \otimes v) := u \cdot v, \quad \text{pour tous } u, v \in \mathcal{A}^*.$$

- $\iota : \mathbb{k} \rightarrow \mathbb{k}\mathcal{A}^*$  est l'application unité définie par

$$\iota : \alpha \longmapsto \alpha\epsilon,$$

pour tout scalaire  $\alpha \in \mathbb{k}$ . (L'élément neutre est le mot vide  $\epsilon$ .)

**Exemple 9.1.2 :**

$$\begin{aligned} \mu.(4abc \otimes (2b + 7cc)) &= 8\mu.(abc \otimes b) + 28\mu.(abc \otimes cc) = 8(abc \cdot b) + 28(abc \cdot cc), \\ \iota(14) &= 14\epsilon. \end{aligned}$$

L'algèbre libre est une  **$\mathbb{k}$ -algèbre graduée connexe** :

- l'espace vectoriel sous-jacent est gradué

$$\mathbb{k}\mathcal{A}^* := \bigoplus_{n \geq 0} \mathbb{k}\mathcal{A}^n, \quad \text{avec } \mathbb{k}\mathcal{A}^n := \text{Vect}_{\mathbb{k}}(\mathcal{A}^n) \quad \text{pour tout } n \geq 0;$$

- le produit de concaténation est homogène :

$$\mu := \bigoplus_{m, n \geq 0} \mu_{m, n} \quad \text{où} \quad \mu_{m, n} = \mathbb{k}\mathcal{A}^m \otimes \mathbb{k}\mathcal{A}^n \rightarrow \mathbb{k}\mathcal{A}^{m+n}.$$

où  $\mu_{m, n}$  est la linéarisation du produit de concaténation restreint aux mots de longueurs  $m$  et  $n$  (respectivement à gauche et à droite de l'opérateur) ;

- le mot vide  $\epsilon$  est l'unique mot de longueur 0 (d'où connexe).

### 9.1.2. Quotient commutatif d'algèbres libres

Soit  $\mathbb{k}\langle \mathcal{A} \rangle := (\mathbb{k}\mathcal{A}^*, \mu, \iota)$  l'algèbre libre sur  $\mathcal{A}$ .

Le **quotient commutatif** de  $\mathbb{k}\langle \mathcal{A} \rangle$  est l'algèbre  $\mathbb{k}[\mathcal{A}]$  des polynômes  $\mathcal{A}$ . Elle peut se définir comme l'espace coinvariant/quotient pour l'action du groupe symétrique sur les mots (1.6). Une présentation équivalente de ce quotient est qu'il s'agit du quotient de  $\mathbb{k}\mathcal{A}^*$  pour la congruence (linéaire)  $ab \equiv ba$ .

**Exemple 9.1.3:** Pour  $\mathcal{A} := \{a, b, c\}$ , dans le quotient commutatif  $\mathbb{k}[\mathcal{A}]$ , on a

$$(abc + 2bcc + ccb) \cdot (a - 2ab) = (cab + 3cbc) \cdot (a - 2ba).$$

On pose  $\theta_{\mathcal{A}}$  la surjection canonique de  $\mathbb{k}\langle \mathcal{A} \rangle$  dans  $\mathbb{k}[\mathcal{A}]$ .

### 9.1.3. Algèbres libres et produit tensoriels complétés

#### Algèbres libres larges

Dans la suite, on considère (la plupart du temps) des alphabets infinis et des combinaisons linéaires infinis de mots. On utilise alors le **complété** de l'algèbre libre (aussi appelé *algèbre libre large*),  $\widehat{\mathbb{k}\langle \mathcal{A} \rangle} := (\widehat{\mathbb{k}\mathcal{A}^*}, \mu, \iota)$  où :

$$\widehat{\mathbb{k}\mathcal{A}^*} := \bigoplus_{n \geq 0} \widehat{\mathbb{k}\mathcal{A}^n},$$

avec  $\widehat{\mathbb{k}\mathcal{A}^n}$  le  $\mathbb{k}$ -espace vectoriel des séries formelles homogènes  $\sum_{|w|=n} c_w w$  de degré  $n$ . Le produit de concaténation linéarisé est encore bien défini et est une application homogène.

**Remarque 9.1.4:** Par abus de notation et de langage, on parlera toujours dans la suite d'algèbre libre, notée  $\mathbb{k}\langle \mathcal{A} \rangle$ , en lieu et place de d'algèbre libre large  $\widehat{\mathbb{k}\langle \mathcal{A} \rangle}$  sans le chapeau.

On procède de même avec le quotient commutatif.

#### Produits tensoriels complétés

De la même manière, on considère le complété du produit tensoriel sur les algèbres libres larges  $\widehat{\mathbb{k}\langle \mathcal{A} \rangle} \widehat{\otimes} \widehat{\mathbb{k}\langle \mathcal{B} \rangle}$ . La composante homogène de degré  $n$  est l'espace des séries formelles :

$$\sum_{\substack{u \in \mathcal{A}^*, v \in \mathcal{B}^* \\ |u|+|v|=n}} c_{u,v} u \otimes v.$$

Le produit  $\bullet := (\mu \widehat{\otimes} \mu) \circ (\text{Id} \widehat{\otimes} \beta \widehat{\otimes} \text{Id})$  est aussi bien défini (voir ci-après la définition de la tresse  $\beta$  (9.8)).

**Remarque 9.1.5:** Comme pour l'algèbre libre large, on oubliera le chapeau sur le produit tensoriel et la notion de complété sera implicite.

## 9.2. Algèbre de Hopf

### 9.2.1. Définition

**Définition 9.2.1 :** Une **algèbre de Hopf** est tuple  $(V, \mu, \iota, \Delta, \varepsilon, S)$  où

- $V$  est un  $\mathbb{k}$ -espace vectoriel,
- le **produit**  $\mu : V \otimes V \rightarrow V$  est une application associative :

$$\mu(\mu(x \otimes y) \otimes z) = \mu(x \otimes \mu(y \otimes z)), \quad (9.1)$$

pour tout  $x, y, z \in V$  ;

- l'**unité**  $\iota : \mathbb{k} \rightarrow V$  :

$$\mu(\iota(\alpha) \otimes x) = \alpha x = \mu(x \otimes \iota(\alpha)), \quad (9.2)$$

pour tout  $x \in V$  et tout  $\alpha \in \mathbb{k}$  ;

- le **coproduit**  $\Delta : V \rightarrow V \otimes V$  est une application co-associative :

$$(\text{Id} \otimes \Delta) \circ \Delta(x) = (\Delta \otimes \text{Id}) \circ \Delta(x), \quad (9.3)$$

pour tout  $x \in V$  ;

- la **counité**  $\varepsilon : V \rightarrow \mathbb{k}$  :

$$(\text{Id} \otimes \varepsilon) \circ \Delta(x) = x = (\varepsilon \otimes \text{Id}) \circ \Delta(x), \quad (9.4)$$

pour tout  $x \in V$  ;

- l'**antipode**  $S : V \rightarrow V$  telle que

$$(S * \text{Id})(x) = \iota(\varepsilon(x)) = (\text{Id} * S)(x), \quad (9.5)$$

pour tout  $x \in V$ , où  $*$  est le **produit de convolution** :

$$f * g := \mu \circ (f \otimes g) \circ \Delta. \quad (9.6)$$

Le produit et le coproduit sont des morphismes l'un pour l'autre :

$$\mu(\Delta(x) \otimes \Delta(y)) = \Delta(\mu(x \otimes y)), \quad (9.7)$$

pour tout  $x, y \in V$ .

### 9.2.2. Axiomes et diagrammes

Les différentes relations de cette définition peuvent se résumer à des diagrammes qui commutent. Les axiomes de produit et d'unité font commuter les diagrammes suivants :

$$\begin{array}{ccc} V \otimes V \otimes V & \xrightarrow{\text{Id} \otimes \mu} & V \otimes V \\ \mu \otimes \text{Id} \downarrow & & \downarrow \mu \\ V \otimes V & \xrightarrow{\mu} & V \end{array} \quad \begin{array}{ccccc} \mathbb{k} \otimes V & \xrightarrow{\iota \otimes \text{Id}} & V \otimes V & \xleftarrow{\text{Id} \otimes \iota} & V \otimes \mathbb{k} \\ & \searrow \simeq & \downarrow \mu & \swarrow \simeq & \\ & & V & & \end{array} \quad (9.1' \text{ et } 9.2')$$

Les axiomes de coproduit et de counité font commuter les mêmes diagrammes où les flèches sont renversées et le produit est remplacé par le coproduit :

$$\begin{array}{ccc}
 V \otimes V \otimes V & \xleftarrow{\text{Id} \otimes \Delta} & V \otimes V \\
 \Delta \otimes \text{Id} \uparrow & & \uparrow \Delta \\
 V \otimes V & \xleftarrow{\Delta} & V
 \end{array}
 \quad
 \begin{array}{ccccc}
 \mathbb{k} \otimes V & \xleftarrow{\varepsilon \otimes \text{Id}} & V \otimes V & \xrightarrow{\text{Id} \otimes \varepsilon} & V \otimes \mathbb{k} \\
 & \searrow \simeq & \uparrow \Delta & \simeq \swarrow & \\
 & & V & & 
 \end{array}
 \quad (9.3' \text{ et } 9.4')$$

L'axiome de l'antipode fait commuter les diagrammes :

$$\begin{array}{ccc}
 V \otimes V & \xrightarrow{\text{Id} \otimes S} & V \otimes V \\
 \Delta \uparrow & & \downarrow \mu \\
 V & \xrightarrow{\varepsilon} \mathbb{k} \xrightarrow{\iota} & V
 \end{array}
 \quad
 \begin{array}{ccc}
 V \otimes V & \xrightarrow{S \otimes \text{Id}} & V \otimes V \\
 \Delta \uparrow & & \downarrow \mu \\
 V & \xrightarrow{\varepsilon} \mathbb{k} \xrightarrow{\iota} & V
 \end{array}
 \quad (9.5')$$

Les propriétés de morphismes entre le produit et le coproduit font commuter le diagramme :

$$\begin{array}{ccc}
 V \otimes V \otimes V \otimes V & \xrightarrow{\text{Id} \otimes \beta \otimes \text{Id}} & V \otimes V \otimes V \otimes V \\
 \Delta \otimes \Delta \uparrow & & \downarrow \mu \otimes \mu \\
 V \otimes V & \xrightarrow{\mu} V \xrightarrow{\Delta} & V \otimes V
 \end{array}
 \quad (9.7')$$

où  $\beta$  est la *tresse* définie comme l'isomorphisme :

$$\beta : a \otimes b \longmapsto b \otimes a. \quad (9.8)$$

La sous-structure  $(V, \mu, \iota)$  est une *algèbre*, la sous-structure  $(V, \Delta, \varepsilon)$  est une *coalgèbre*, et si on oublie seulement l'antipode, la sous-structure  $(V, \mu, \iota, \Delta, \varepsilon)$  est une *bialgèbre*.

**Exemple 9.2.2 :** Soit  $\mathcal{A}$  l'ensemble  $\{a, b, c\}$ . On considère le  $\mathbb{R}$ -espace vectoriel engendré par l'ensemble des mots  $\mathcal{A}^*$ , noté  $\mathbb{R}\mathcal{A}^*$ .

$$\pi abc + abb + 4cbbbc + 5 \in \mathbb{R}\mathcal{A}^*$$

Le produit est le produit de concaténation (linéarisé) :

$$(ab + 3c) \cdot (cab + aab - 4a) = abcab + abaab - 4aba + 3ccab + 3caab - 12ca,$$

et l'application unité  $\iota$  définit par :

$$\iota : \begin{cases} \mathbb{k} & \rightarrow \mathbb{k}\mathcal{A}^* \\ \alpha & \mapsto \alpha \varepsilon \end{cases}.$$

Par exemple,

$$\iota(-3) = -3\varepsilon.$$

Cette structure d'algèbre correspond à l'algèbre libre  $\mathbb{k}\langle \mathcal{A} \rangle$ .

On définit le coproduit sur les symboles de  $\mathcal{A}$  et la définition s'étend naturellement à partir du morphisme avec le produit (9.7) :

$$\Delta(a) := \epsilon \otimes a + a \otimes \epsilon, \quad \text{pour tout } a \in \mathcal{A},$$

d'où

$$\begin{aligned} \Delta(abc) &= \Delta(a) \bullet \Delta(b) \bullet \Delta(c), \\ &= (\epsilon \otimes a + a \otimes \epsilon) \bullet (\epsilon \otimes b + b \otimes \epsilon) \bullet (\epsilon \otimes c + c \otimes \epsilon), \\ &= \epsilon \otimes abc + b \otimes ac + c \otimes ab + bc \otimes a + a \otimes bc + ab \otimes c + ac \otimes b + abc \otimes \epsilon \end{aligned}$$

avec  $\bullet := (\mu \otimes \mu) \circ (\text{Id} \otimes \beta \otimes \text{Id})$ . Une autre manière de définir le coproduit est :

$$\Delta(w) = \sum_{u,v} \langle w \mid u \sqcup v \rangle u \otimes v,$$

où  $\langle x \mid y \rangle := \delta_{x,y}$  (avec  $\delta_{x,y}$  le symbole de Kronecker) et  $\sqcup$  est le produit de mélange (1.5).

À partir de l'axiome (9.5), on déduit l'unique antipode :

$$\mathcal{S}(w) := (-1)^{|w|} w^\sim.$$

Par exemple, on a

$$\begin{aligned} (\mathcal{S} * \text{Id})(\epsilon) &= \mathcal{S}(\epsilon) \cdot \epsilon = \mathcal{S}(\epsilon) = \epsilon \\ (\mathcal{S} * \text{Id})(a) &= \mathcal{S}(a) \cdot \epsilon + \mathcal{S}(\epsilon) \cdot a = 0 \\ &\Rightarrow \mathcal{S}(a) = -a \\ (\mathcal{S} * \text{Id})(ab) &= \mathcal{S}(\epsilon) \cdot ab + \mathcal{S}(a) \cdot b + \mathcal{S}(b) \cdot a + \mathcal{S}(ab) \cdot \epsilon = 0 \\ &= ab - ab - ba + \mathcal{S}(ab) \\ &\Rightarrow \mathcal{S}(ab) = ba \\ (\mathcal{S} * \text{Id})(abc) &= \iota(\varepsilon(abc)) = 0 \\ &= \mathcal{S}(\epsilon) \cdot abc + \mathcal{S}(a) \cdot bc + \mathcal{S}(b) \cdot ac + \mathcal{S}(c) \cdot ab + \\ &\quad \mathcal{S}(ab) \cdot c + \mathcal{S}(bc) \cdot a + \mathcal{S}(ac) \cdot b + \mathcal{S}(abc) \cdot \epsilon = 0 \\ &= abc - abc - bac - cab + bac + cba + cab + \mathcal{S}(abc) \\ &\Rightarrow \mathcal{S}(abc) = -cba \end{aligned}$$

**Remarque 9.2.3 :** Il est souvent agréable de décrire le coproduit par la notation de Sweedler :

$$\Delta(x) = \sum x^L \otimes x^R.$$

### 9.2.3. Algèbre de Hopf combinatoire

Dans la suite, on s'intéressera plus particulièrement au cas où la/les base(s) de l'espace vectoriel sous-jacent est/sont indexée(s) par des structures combinatoires.

**Définition 9.2.4 :** Une *algèbre de Hopf combinatoire* est une algèbre de Hopf où l'espace vectoriel sous-jacent  $V$  à pour base une classe combinatoire  $\mathcal{C}$  :

$$V = \text{Vect}_{\mathbb{k}}(\mathcal{C}) := \bigoplus_{n \geq 0} \text{Vect}_{\mathbb{k}}(\mathcal{C}_n).$$

**Remarque 9.2.5 :** Cette définition d'algèbre de Hopf « combinatoire » n'est pas universelle. Dans la littérature, on peut trouver différentes définitions dont l'unique point commun est l'utilisation d'algèbres de Hopf dans un cadre combinatoire. Cela étant, dans ce manuscrit, une algèbre de Hopf combinatoire est une algèbre de Hopf indexée par une classe combinatoire.

Dans l'introduction de cette partie ainsi que les précédents chapitres de ce manuscrit, les algèbres de Hopf présentées sont toutes des algèbres de Hopf dites *combinatoires*. L'algèbre des fonctions symétriques [Mac98] est indexée par les partitions d'entiers (§ 1.1.1), l'algèbre des fonctions symétriques non-commutatives [Gel+95] est indexée par les compositions d'entiers (§ 1.1.3) et l'algèbre des fonctions quasi-symétriques libres [MR95] est indexée par les permutations (§ 1.1.5).

La série de Hilbert des dimensions de l'espace vectoriel d'une algèbre de Hopf combinatoire est égale à la série génératrice de la classe combinatoire  $\mathcal{C}$ .

### Bialgèbre graduée connexe et antipode

Dans le cas d'une bialgèbre graduée connexe, la structure est automatiquement enrichie d'une antipode (unique) :

**Proposition 9.2.6 ([MM65, Proposition 8.2]) :** Une bialgèbre  $(V, \mu, \iota, \Delta, \varepsilon)$  qui est graduée et connexe, est automatiquement munie d'une structure d'algèbre de Hopf avec l'antipode définie récursivement par :

$$S(x) = \begin{cases} x & \text{si } \deg(x) = 0, \\ \sum_{x^L \neq x} \mu(S(x^L) \otimes x^R) & \text{sinon.} \end{cases}$$

Il existe d'autres formulations de l'antipode, on peut notamment citer celle donnée par Mitsuhiro TAKEUCHI [Tak71, Lemme 14].



## 9.3. Théorie des catégories et catégorie magmatique

La *théorie des catégories* permet l'étude des structures mathématiques et des relations entre elles. Cette théorie a été développée dans les années quarante par Samuele EILENBERG et Saunders MAC LANE puis fut propagée dans les années soixante par Alexander GROTHENDIECK.

Elle est utilisée dans de nombreux domaines des mathématiques et émerge dans nombre de domaines informatiques : *logique*, *base de données* ou encore *combinatoire algébrique*. En combinatoire, l'intérêt de cette théorie est de fournir un cadre *fortement typé*, une démarche *constructive* et surtout de dissocier l'aspect *algébrique*, de l'aspect purement *combinatoire*. Dans cette partie, nous utiliserons la théorie des catégories pour définir la notion de *type d'alphabet*. Cette notion est l'un des deux nœuds centraux de la *théorie des réalisations polynomiales d'algèbres de Hopf* [HP15].

### 9.3.1. Catégories

Pour définir la notion de *catégorie*, on devrait tout d'abord définir une structure plus générale que la notion d'*ensembles* : les *classes*. Dans la mesure où dans la suite nous n'utilisons que des ensembles, on dira simplement qu'une *classe* est une collection d'objets qui ne souffre pas du paradoxe de Bertrand A. W. RUSSELL : on peut parler de la classe des ensembles qui ne s'appartiennent pas eux-mêmes.

**Définition 9.3.1 :** Une *catégorie*  $\mathbb{C}$  se compose de deux *classes* :

- une classe d'*objets*, notée  $Ob(\mathbb{C})$  ;
- une classe de *morphismes* (ou *flèches*)  $f : A \rightarrow B$  (avec  $A, B \in Ob(\mathbb{C})$ ) entre les objets de  $\mathbb{C}$ , notée  $Hom(\mathbb{C})$  ;

ainsi que d'une loi de *composition* de morphismes  $\circ$  : pour tous objets  $A, B, C \in Ob(\mathbb{C})$  et tous morphismes  $f : A \rightarrow B$  et  $g : B \rightarrow C$ , la loi associe  $g \circ f : A \rightarrow C$  ; telle que les axiomes suivants sont satisfaits :

$$h \circ (g \circ f) = (h \circ g) \circ f \quad \text{[Associativité]}$$

pour tous  $f : A \rightarrow B$ ,  $g : B \rightarrow C$  et  $h : C \rightarrow D$ , et

$$Id_B \circ f = f = f \circ Id_A \quad \text{[Identité]}$$

pour tout  $f : A \rightarrow B$ , où  $Id_A$  et  $Id_B$  sont les morphismes identités respectivement de  $A$  et  $B$ .

On note  $Hom_{\mathbb{C}}(A, B)$  les morphismes  $f$  de  $A$  vers  $B$ , et par abus de notation, on va noter simplement  $A \in \mathbb{C}$  pour signifier que  $A$  est un objet de  $\mathbb{C}$  (au lieu de  $A \in Ob(\mathbb{C})$ ).

**Exemple 9.3.2 :** Quelques catégories :

- **Ens** la catégorie des *ensembles* où les objets sont les ensembles et les morphismes sont toutes les fonctions (totales) d'un ensemble à un autre ;

- **Rel** la catégorie des *relations* où les objets sont les ensembles et les morphismes sont les relations binaires ;
- **Mon** la catégorie des *monoïdes* où les objets sont les monoïdes et les morphismes sont les homomorphismes de monoïdes.

### 9.3.2. Foncteurs

**Définition 9.3.3 :** Un *foncteur*  $\mathcal{F} : \mathbb{C} \rightarrow \mathbb{D}$  de la catégorie  $\mathbb{C}$  vers la catégorie  $\mathbb{D}$  est la donnée d'une fonction qui

- à tout objet  $A$  de  $\mathbb{C}$  associe un objet  $\mathcal{F}(A)$  de  $\mathbb{D}$ ,
- à tout morphisme  $f : A \rightarrow B$  de  $\mathbb{C}$  associe un morphisme  $\mathcal{F}(f) : \mathcal{F}(A) \rightarrow \mathcal{F}(B)$  tel que
- pour tout  $A \in \text{Ob}(\mathbb{C})$ ,

$$\mathcal{F}(Id_A) = Id_{\mathcal{F}(A)} ,$$

- pour tous  $f : A \rightarrow B$  et  $g : B \rightarrow C$  deux morphismes de  $\mathbb{C}$ ,

$$\mathcal{F}(g \circ f) = \mathcal{F}(g) \circ \mathcal{F}(f) .$$

**Exemple 9.3.4 :** L'étoile de Kleene associant à tout ensemble  $A$  le monoïde libre :

$$A^* = \{\epsilon\} \cup A \cup (A \times A) \cup (A \times A \times A) \cup \dots$$

et à tout morphisme  $f : A \rightarrow B$ , l'application :

$$f^* : \begin{cases} A^* \rightarrow B^* , \\ (a_1, \dots, a_k) \mapsto (i(a_1), \dots, i(a_k)) . \end{cases}$$

est un *foncteur*  $(\_)^*$  de la catégorie des ensembles **Ens** vers la catégorie des monoïdes (gradués) **Mon** (ou plus précisément **gMon**).

**Exemple 9.3.5 :** Soit  $\mathbb{k}$  un anneau/corps. La construction associant à tout monoïde  $M$ , l'algèbre du monoïde  $\mathbb{k}M$  est une construction fonctorielle.

### 9.3.3. Transformations naturelles

Une *transformation naturelle* est une généralisation de la notion de morphisme au niveau des catégories sur les foncteurs.

**Définition 9.3.6 :** Soient  $\mathcal{F}$  et  $\mathcal{G}$  deux foncteurs de la catégorie  $\mathbb{C}$  vers la catégorie  $\mathbb{D}$ . Une *transformation naturelle*  $\eta$  de  $\mathcal{F}$  à  $\mathcal{G}$  est une donnée associant  $\eta_A : \mathcal{F}(A) \rightarrow \mathcal{G}(A)$  à tout objet  $A$  de  $\mathbb{C}$  telle que pour tout morphisme  $f : A \rightarrow B$  (avec  $B$  un objet de  $\mathbb{C}$ ),

$$\mathcal{G}(f) \circ \eta_A = \eta_B \circ \mathcal{F}(f) .$$

Cette définition se résume par la commutation du diagramme suivant :

$$\begin{array}{ccc} \mathcal{F}(A) & \xrightarrow{\mathcal{F}(f)} & \mathcal{F}(B) \\ \eta_A \downarrow & & \downarrow \eta_B \\ \mathcal{G}(A) & \xrightarrow{\mathcal{G}(f)} & \mathcal{G}(B) \end{array}$$

### 9.3.4. Catégorie magmatique

**Définition 9.3.7 :** Une **catégorie magmatique** (ou *catégorie monoïdale non-unitaire*) est une catégorie  $\mathbb{C}$  munie de

- un **bifoncteur**  $\bullet : \mathbb{C} \times \mathbb{C} \rightarrow \mathbb{C}$ ,
- une transformation naturelle  $\alpha$  telle que pour tous objets  $A, B, C \in \mathbb{C}$ , le morphisme  $\alpha_{A,B,C}$  de  $(A \bullet B) \bullet C$  vers  $A \bullet (B \bullet C)$  est un isomorphisme, tel que le diagramme suivant commute

$$\begin{array}{ccccc} & & (A \bullet (B \bullet C)) \bullet D & & \\ & \nearrow \alpha_{A,B,C} \bullet Id_D & & \searrow \alpha_{A,B \bullet C,D} & \\ ((A \bullet B) \bullet C) \bullet D & & & & A \bullet ((B \bullet C) \bullet D) \\ & \searrow \alpha_{A \bullet B,C,D} & & \swarrow Id_A \bullet \alpha_{B,C,D} & \\ & (A \bullet B) \bullet (C \bullet D) & \xrightarrow{\alpha_{A,B,C \bullet D}} & A \bullet (B \bullet (C \bullet D)) & \end{array}$$

On parle de **catégorie monoïdale** lorsqu'il existe dans  $\mathbb{C}$

- un objet  $I \in \mathbb{C}$ ,
- de deux transformations naturelles  $\eta, v$  telles que pour tout objet  $A$ , les morphismes  $\eta_A : I \bullet A \rightarrow A$  et  $v_A : A \bullet I \rightarrow A$  sont des isomorphismes, tel que le diagramme suivant commute

$$\begin{array}{ccc} (A \bullet I) \bullet B & \xrightarrow{\alpha_{A,I,B}} & A \bullet (I \bullet B) \\ & \searrow v_A \bullet Id_B \quad \swarrow Id_A \bullet \eta_B & \\ & A \bullet B & \end{array}$$

L'espace  $A \bullet B$  fait référence au produit tensoriel de  $A$  et  $B$  (et  $I$  à l'**élément neutre** de  $\mathbb{C}$ ). La transformation naturelle  $\alpha$  est l'**associateur**. (Les transformations  $\eta$  et  $v$  sont appelées **contraintes d'identité**.)

**Exemple 9.3.8 :** La catégorie des  $\mathbb{k}$ -espaces vectoriels munie de  $\otimes$ , le produit tensoriel usuel, est une **catégorie monoïdale** (où  $\mathbb{k}$  est l'élément neutre de la catégorie).

Une *catégorie magmatique/monoïdale tressée* est une catégorie magmatique/monoïdale  $(\mathbb{C}, \bullet)$  munie de la transformation naturelle  $\beta$ , appelé *volte*, où pour tout  $A, B \in \mathbb{C}$ , le morphisme  $\beta_{A,B} : A \bullet B \xrightarrow{\sim} B \bullet A$  est un isomorphisme naturel faisant commuter les diagrammes suivants :

$$\begin{array}{ccc}
 & B \bullet A \bullet C & \\
 \beta_{A,B} \bullet Id_C \nearrow & & \searrow Id_B \bullet \beta_{A,C} \\
 A \bullet B \bullet C & \xrightarrow{\beta_{A,B \bullet C}} & B \bullet C \bullet A
 \end{array}
 \qquad
 \begin{array}{ccc}
 & A \bullet C \bullet B & \\
 Id_A \bullet \beta_{B,C} \nearrow & & \searrow \beta_{A,C} \bullet Id_B \\
 A \bullet B \bullet C & \xrightarrow{\beta_{A \bullet B, C}} & C \bullet A \bullet B
 \end{array}$$

Une catégorie magmatique/monoïdale tressée est *symétrique* si la volte  $\beta$  satisfait  $\beta^2 = Id$ . On dit que  $\beta$  est une *symétrie*.

# 10. Réalisation polynomiale d'algèbre de Hopf combinatoire

## 10.1. Type d'alphabet

Dans l'introduction, on donne l'exemple de la réalisation polynomiale de l'algèbre de Hopf **FQSym**, et l'on calcule très facilement l'inverse de l'exponentielle, définie dans **FQSym**, à partir de la réalisation polynomiale. Dans la même idée de simplification, vérifier les axiomes de coproduit et de compatibilités avec le produit n'est pas évident. Les articles [Gel+95 ; DHT02] introduisent l'*astuce de dédoublement d'alphabet* qui s'inspire de la propriété classique de l'exponentielle (encore une fois) sur deux variables :

$$\exp(x + y) = \exp(x) \exp(y).$$

La technique de dédoublement d'alphabet étend l'idée évoquée dans [Mor82] pour expliquer le coproduit. L'idée consiste à définir l'opération de coproduit comme une opération « simple » sur la réalisation polynomiale. On considère l'alphabet  $\mathcal{A}$  de réalisation comme une « somme » de deux autres,  $\mathcal{B} \oplus \mathcal{C}$ , et on extrait les sous-mots issus de  $\mathcal{B}$  d'un côté et de  $\mathcal{C}$  d'un autre :

$$\Delta(f(\mathcal{B} \oplus \mathcal{C})) = \sum w|_{\mathcal{B}} \otimes w|_{\mathcal{C}} = \sum f^L(\mathcal{B}) \otimes f^R(\mathcal{C}).$$

Afin de définir cette astuce, on regroupe ces alphabets de réalisations au sein d'une catégorie magmatique où la somme d'alphabets définit le bifoncteur (§ 9.3.4) :

**Définition 10.1.1 :** Un *type d'alphabet* **Alph** est une (petite) catégorie magmatique d'ensembles  $(\mathbb{C}, \oplus)$  où toute *somme d'alphabets*  $\mathcal{A} \oplus \mathcal{B}$  est munie de deux morphismes injectifs (*gauche* et *droite*) :

$$g_{\mathcal{A}, \mathcal{B}} : \begin{cases} \mathcal{A} & \longrightarrow & \mathcal{A} \oplus \mathcal{B} \\ a & \longmapsto & a \end{cases} \quad \text{et} \quad d_{\mathcal{A}, \mathcal{B}} : \begin{cases} \mathcal{B} & \longrightarrow & \mathcal{A} \oplus \mathcal{B} \\ b & \longmapsto & b \end{cases}.$$

et où l'ensemble sous-jacent de  $\mathcal{A} \oplus \mathcal{B}$  est l'union disjointe  $\mathcal{A} \sqcup \mathcal{B}$ .

**Remarque 10.1.2 :** La notion de catégorie est utilisée non pas pour l'étude des morphismes, mais pour définir aisément une collection d'ensembles ayant la même structure.

Moins formellement, cette définition correspond à

**Définition 10.1.3 :** Un **type d'alphabet** **Alph** est défini par les données suivantes :

- une notion d'ensembles  $\mathcal{A}$  munis d'une certaine structure  $R$  : les **alphabets**,
- une notion de morphisme contenant l'identité et stable par composition,
- une opération associative à isomorphisme près sur les alphabets qui associe, à tout couple d'alphabets  $(\mathcal{A}, R)$  et  $(\mathcal{B}, S)$ , un nouvel alphabet  $(\mathcal{A} \oplus \mathcal{B}, T)$  défini comme étant l'union disjointe de  $\mathcal{A}$  et  $\mathcal{B}$  munie d'une structure  $T$  telle que  $R$  (respectivement  $S$ ) est la restriction de  $T$  à l'alphabet  $\mathcal{A}$  (respectivement à  $\mathcal{B}$ ) : la **somme d'alphabets**,
- ainsi que deux injections  $g, d$  (gauche et droite) qui associent à tous alphabets  $\mathcal{A}$  et  $\mathcal{B}$  des morphismes injectifs canoniques :

$$g_{\mathcal{A}, \mathcal{B}} : \left\{ \begin{array}{ccc} \mathcal{A} & \longrightarrow & \mathcal{A} \oplus \mathcal{B} \\ a & \longmapsto & a \end{array} \right. , \quad \text{et} \quad d_{\mathcal{A}, \mathcal{B}} : \left\{ \begin{array}{ccc} \mathcal{B} & \longrightarrow & \mathcal{A} \oplus \mathcal{B} \\ b & \longmapsto & b \end{array} \right. .$$

### 10.1.1. Exemples

Voici quelques exemples de **types d'alphabets** que nous utilisons par la suite :

#### Alphabets nus

Soit **Ens** la catégorie des ensembles (tout nus, sans structure particulière) et soit  $\oplus$  le bifoncteur du carré tensoriel **Ens**  $\times$  **Ens** dans **Ens** défini comme l'union disjointe des ensembles :

$$\mathcal{A} \oplus \mathcal{B} := \mathcal{A} \sqcup \mathcal{B} \quad \text{pour tous } \mathcal{A}, \mathcal{B} \in \mathbf{Ens},$$

et la somme des morphismes :

$$f \oplus h : \left\{ \begin{array}{ccc} \mathcal{A} \oplus \mathcal{B} & \longrightarrow & \mathcal{C} \oplus \mathcal{D} \\ a \in \mathcal{A} & \longmapsto & g_{\mathcal{C}, \mathcal{D}} \circ f(x), \\ b \in \mathcal{B} & \longmapsto & d_{\mathcal{C}, \mathcal{D}} \circ h(x), \end{array} \right.$$

pour tous morphismes  $f : \mathcal{A} \rightarrow \mathcal{C}$  et  $h : \mathcal{B} \rightarrow \mathcal{D}$ , avec gauche  $g$  et droite  $d$  respectivement les morphismes injectifs canoniques. Cette première **catégorie magmatique**  $(\mathbf{Ens}, \oplus)$  définit le **type des alphabets nus**.

**Remarque 10.1.4 :** Il s'agit même d'une catégorie monoïdale où l'ensemble vide est l'élément neutre.

Dans la suite, on va principalement s'intéresser aux alphabets infinis. En se restreignant à la sous-catégorie  $\mathbf{Ens}^\infty$  des ensembles infinis, on obtient le (sous-) **type des alphabets nus infinis**.

On utilisera ce type d'alphabet tout au long des prochaines sections. On l'utilisera pour construire l'algèbre de Hopf **WSym** [Wol36 ; RS06] au travers de différents exemples accompagnant les définitions de réalisations polynomiales d'algèbres de Hopf.

## Alphabets totalement ordonnés

Soit  $\mathbf{Ens}_\infty^\infty$  la catégorie des ensembles infinis totalement ordonnés. On définit la somme d'alphabets  $\oplus$  comme le bifoncteur :

$$(\mathcal{A}, <_{\mathcal{A}}) \oplus (\mathcal{B}, <_{\mathcal{B}}) := (\mathcal{A} \sqcup \mathcal{B}, <_{\mathcal{A} \oplus \mathcal{B}})$$

où  $<_{\mathcal{A} \oplus \mathcal{B}}$  est défini par

$$x <_{\mathcal{A} \oplus \mathcal{B}} y \iff \begin{cases} x <_{\mathcal{A}} y & \text{avec } x, y \in \mathcal{A}, \\ x \in \mathcal{A} & \text{et } y \in \mathcal{B}, \\ x <_{\mathcal{B}} y & \text{avec } x, y \in \mathcal{B}, \end{cases}$$

pour tous alphabets  $\mathcal{A}$  et  $\mathcal{B}$ , et est défini comme précédemment pour tous morphismes. On nomme cette catégorie magmatique le *type des alphabets (infinis) totalement ordonnés*.

Ce type permettra de définir les réalisations des algèbres de Hopf **FQSym** ([DHT02 ; Duc+11b] et § 12.1) et **WQSym** [NT06].

## Alphabets totalement ordonnés avec différences

On considère la sous-catégorie de  $\mathbf{Ens}_\infty^\infty$  des ensembles totalement ordonnés  $(\mathcal{A}, <_{\mathcal{A}})$  munie d'une application  $D_{\mathcal{A}} : \mathcal{A} \times \mathcal{A} \rightarrow \mathbb{Z}$  (où  $\mathbb{Z} := \mathbb{Z} \cup \{-\infty, +\infty\}$ ) telle que :

- $D_{\mathcal{A}}(x, y) = 0$  si et seulement si  $x = y$ ,
- $D_{\mathcal{A}}(x, y) > 0$  si  $y <_{\mathcal{A}} x$  et  $x \neq y$ ,
- $D_{\mathcal{A}}(x, y) = -D_{\mathcal{A}}(y, x)$ ,
- $D_{\mathcal{A}}(x, z) = D_{\mathcal{A}}(x, y) + D_{\mathcal{A}}(y, z)$ .

On construit la somme d'alphabets  $\mathcal{A} \oplus \mathcal{B}$  comme précédemment en enrichissant la structure d'une application  $D_{\mathcal{A} \oplus \mathcal{B}}$  :

$$D_{\mathcal{A} \oplus \mathcal{B}}(x, y) := \begin{cases} D_{\mathcal{A}}(x, y) & \text{avec } x, y \in \mathcal{A}, \\ -\infty & \text{si } x \in \mathcal{A} \text{ et } y \in \mathcal{B}, \\ +\infty & \text{si } x \in \mathcal{B} \text{ et } y \in \mathcal{A}, \\ D_{\mathcal{B}}(x, y) & \text{avec } x, y \in \mathcal{B}. \end{cases}$$

Cette dernière catégorie est le *type des alphabets avec différences*, noté  $\mathbf{Ens}_D^\infty$ . Il va permettre de construire l'algèbre de Hopf **PQSym** (§ 12.2).

### 10.1.2. Opérations

Les différents *types d'alphabets* de la précédente sous-section vont permettre de définir les réalisations polynomiales de nombreuses algèbres de Hopf : **WSym** avec le type des alphabets nus, **FQSym**, **WQSym** avec le type des alphabets totalement ordonnés, **PQSym** avec les alphabets totalement ordonnés avec différences.

D'autres algèbres de Hopf peuvent se définir simplement en enrichissant ou en couplant les réalisations des algèbres de Hopf précédentes. Par exemple, pour définir la réalisation polynomiale de l'algèbre des fonctions quasi-symétriques de niveau  $l$  arbitraire, **FQSym**<sup>(1)</sup> [NT04b], on va utiliser la même application partitionnante que pour réaliser **FQSym**, la standardisation, enrichie d'une notion de couleur. Pour réaliser cette algèbre, on souhaite donc manipuler des alphabets totalement ordonnés *colorés* par un ensemble fini  $K$ . Il en va de même pour les algèbres **FQSym**, **WSym**, **PQSym** signées ou encore l'algèbre de Hopf cambrienne [CP15]. D'autres réalisations d'algèbres de Hopf nécessitent de manipuler des alphabets de *bilettres*, par exemple **BWQSym** [Bul+13].

Pour ces différents cas, on introduit dans cette sous-section les opérations de *coloration* et de *produit cartésien* des types d'alphabets.

### Coloration

Soit  $K$  un ensemble et soit **Alph** un type d'alphabet. On définit la coloration de **Alph** =  $(\mathbb{C}, \oplus)$  par l'ensemble  $K$ , notée  $\mathbf{Alph}^K := (\mathbb{D}, \oplus_K)$ , comme le type d'alphabet défini par la catégorie magmatique :  $(\mathbb{D}, \oplus_K)$  où les ensembles/alphabets sont les produits cartésiens  $\mathcal{A} \times K$  des ensembles  $\mathcal{A}$  de  $\mathbb{C}$  avec  $K$ , les morphismes sont les applications  $(f, Id_K)$  avec  $f$  un morphisme de  $\mathbb{C}$  et  $Id_K$  l'identité pour l'ensemble des couleurs  $K$ , et, la somme d'alphabets est définie par :

$$(\mathcal{A} \times K) \oplus_K (\mathcal{B} \times K) := (\mathcal{A} \oplus \mathcal{B}) \times K.$$

### Produit cartésien

Soient **Alph** :=  $(\mathbb{C}, \oplus)$  et **Alph'** :=  $(\mathbb{C}', \oplus')$  deux types d'alphabets.

On note **Alph** × **Alph'** le produit cartésien des types d'alphabets et on enrichit cette catégorie d'une structure de type d'alphabet avec la somme d'alphabet  $\oplus_\times$  :

$$(\mathcal{A} \times \mathcal{A}') \oplus_\times (\mathcal{B} \times \mathcal{B}') := (\mathcal{A} \oplus \mathcal{B}) \times (\mathcal{A}' \oplus' \mathcal{B}').$$

pour tous couples alphabets  $\mathcal{A} \times \mathcal{A}'$  et  $\mathcal{B} \times \mathcal{B}'$  de **Alph** × **Alph'**, et :

$$(f, f') \oplus_\times (h, h') := ((f \oplus h), (f' \oplus' h')) .$$

pour tous couples de morphismes  $(f, f')$  et  $(h, h')$  de **Alph** × **Alph'**.

## 10.2. Application partitionnante

Dans cette section, on définit la notion d'*application partitionnante*. Les différentes propriétés de ces applications vont permettre de construire automatiquement une algèbre de Hopf combinatoire dans la section suivante. Cette notion permet de formaliser des propriétés d'indépendance vis-à-vis de l'alphabet des applications comme



la standardisation, le tassement ou la parkisation qui permettent de définir respectivement **FQSym**, **WQSym** et **PQSym**. Ces bonnes propriétés se traduisent par des conditions combinatoires élémentaires sur les mots.

Dans les deux sections suivantes, on définit des constructions automatiques d'algèbres de Hopf (en dualité) à partir d'une application partitionnante  $\phi$ .

**Remarque 10.2.1 :** Afin de minimiser les notations, on s'autorise à confondre les notations  $d_{\mathcal{A},\mathcal{B}}$  et  $g_{\mathcal{A},\mathcal{B}}$  des morphismes injectifs droit et gauche et leurs extensions aux mots :

$$d_{\mathcal{A},\mathcal{B}} : a_1 a_2 \cdots a_m \longmapsto d_{\mathcal{A},\mathcal{B}}(a_1) d_{\mathcal{A},\mathcal{B}}(a_2) \cdots d_{\mathcal{A},\mathcal{B}}(a_m)$$

et respectivement avec  $g_{\mathcal{A},\mathcal{B}}$ .

**Définition 10.2.2 :** Soit **Alph** un type d'alphabet et soit  $\mathcal{C}$  une classe combinatoire.

Une **Alph-application partitionnante** est une donnée  $\phi$  qui associe, à tout alphabet  $\mathcal{A}$  de **Alph**, une application  $\phi_{\mathcal{A}} : \mathcal{A}^* \rightarrow \mathcal{C}$  telle que

1.  $\phi_{\mathcal{A}}$  est surjective ;
2.  $\phi_{\mathcal{A}}$  est graduée :  $|\phi_{\mathcal{A}}(w)| = |w|$  ;
3.  $\phi$  commute avec les morphismes gauche et droite :

$$\phi_{\mathcal{A}}(w) = \phi_{\mathcal{A} \oplus \mathcal{B}}(g_{\mathcal{A},\mathcal{B}}(w)) \quad \text{et} \quad \phi_{\mathcal{A}}(w) = \phi_{\mathcal{B} \oplus \mathcal{A}}(d_{\mathcal{B},\mathcal{A}}(w)) ,$$

pour tout alphabet  $\mathcal{B}$  ;

4.  $\phi_{\mathcal{A}}$  est compatible aux facteurs :

$$\text{si } \phi_{\mathcal{A}}(uv) = \phi_{\mathcal{A}}(u'v') \quad \text{alors} \quad \begin{cases} \phi_{\mathcal{A}}(u) = \phi_{\mathcal{A}}(u') \\ \phi_{\mathcal{A}}(v) = \phi_{\mathcal{A}}(v') \end{cases} ,$$

pour tous  $u, u' \in \mathcal{A}^m$  et tous  $v, v' \in \mathcal{A}^n$  ;

5.  $\phi$  est compatible au mélange :

$$\text{si } \left\{ \begin{array}{l} \phi_{\mathcal{A}}(u) = \phi_{\mathcal{A}}(u') \\ \phi_{\mathcal{B}}(v) = \phi_{\mathcal{B}}(v') \end{array} \right\} \quad \text{alors} \quad \phi_{\mathcal{A} \oplus \mathcal{B}}((uv) \cdot \sigma) = \phi_{\mathcal{A} \oplus \mathcal{B}}((u'v') \cdot \sigma) ,$$

pour tous mots  $u, u' \in \mathcal{A}^m$ , tous mots  $v, v' \in \mathcal{B}^n$  et toute permutation  $\sigma \in \text{Sh}_{m,n}$ .

On dit que  $\phi$  est une **Alph-application partitionnante** sur la classe combinatoire  $\mathcal{C}$ .

**Remarque 10.2.3 :** Dans l'expression  $\phi_{\mathcal{A} \oplus \mathcal{B}}((uv) \cdot \sigma)$ , on utilise implicitement les transformations gauche et droite :

$$\phi_{\mathcal{A} \oplus \mathcal{B}}((uv) \cdot \sigma) := \phi_{\mathcal{A} \oplus \mathcal{B}}((g_{\mathcal{A},\mathcal{B}}(u) d_{\mathcal{A},\mathcal{B}}(v)) \cdot \sigma) .$$

**Exemple 10.2.4 :** On considère la classe combinatoire des partitions d'ensembles  $\mathcal{C}_{\text{Par}}$  (cf. notation (1.15)) et le type des alphabets nus infinis  $\mathbf{Ens}^\infty$ . La donnée  $\text{occ}$  qui associe à chaque alphabet nu infini  $\mathcal{A}$  :

$$\text{occ}_{\mathcal{A}} : \begin{cases} \mathcal{A}^* & \longrightarrow \mathcal{C}_{\text{Par}} \\ w := w_1 w_2 \cdots w_k & \longmapsto (\{p \in [k] \mid a_p = l\})_{l \in \mathcal{A}(w)} \end{cases}$$

est une  $\mathbf{Ens}^\infty$ -application partitionnante. (On rappelle que  $\mathcal{A}(w)$  est la restriction de l'ensemble  $\mathcal{A}$  aux symboles présents dans  $w$ .)

$$\text{occ}_{\mathcal{A}}(\text{tempeste}) = \{257 \mid 16 \mid 3 \mid 4\}.$$

Comme  $\mathcal{A}$  est infini,  $\text{occ}_{\mathcal{A}}$  est surjective (Condition 1) (il suffit de se fixer une injection de  $\mathcal{A}$  dans les entiers pour montrer qu'à toute partition on peut associer au moins un mot). Par définition de  $\text{occ}$ , chaque mot de longueur  $n$  est associé à une partition de l'ensemble  $[n]$  donc  $\text{occ}_{\mathcal{A}}$  est graduée (Condition 2). Par définition du type d'alphabet, on déduit que  $\text{occ}$  commute avec les morphismes gauche et droit (Condition 3). La compatibilité aux facteurs (Condition 4) est aussi satisfaite. En effet, les partitions de  $\text{occ}_{\mathcal{A}}(u)$  et  $\text{occ}_{\mathcal{A}}(v)$  sont incluses dans  $\text{occ}_{\mathcal{A}}(uv)$  : si  $u \in \mathcal{A}^m$  alors

$$\begin{aligned} \text{occ}_{\mathcal{A}}(u) &= \{\{i \mid i \in S \text{ et } i \leq m\} \mid S \in \text{occ}_{\mathcal{A}}(uv)\} \setminus \{\emptyset\}, \\ \text{occ}_{\mathcal{A}}(v) &= \{\{i - m \mid i \in S \text{ et } i > m\} \mid S \in \text{occ}_{\mathcal{A}}(uv)\} \setminus \{\emptyset\}. \end{aligned}$$

Enfin pour montrer la compatibilité au mélange (Condition 5), on utilise les deux propriétés suivantes : si  $u \in \mathcal{A}^m$  et  $v \in \mathcal{B}^n$  alors

$$\text{occ}_{\mathcal{A} \oplus \mathcal{B}}(uv) = \text{occ}_{\mathcal{A}}(u) \cup \{\{i + m \mid m \in S\} \mid S \in \text{occ}_{\mathcal{B}}(v)\}$$

et l'action des permutations sur les mots commute avec  $\text{occ}$  :

$$\text{occ}_{\mathcal{A}}(u \cdot \sigma) = \sigma \cdot \text{occ}_{\mathcal{A}}(u) := \text{Par}[\sigma](\text{occ}_{\mathcal{A}}(u)),$$

pour toute permutation  $\sigma$ . (cf. transport de structures des partitions, Exemple 1.5.6).

**Exemple 10.2.5 :** On considère  $\mathcal{C} := \bigsqcup_{n \geq 0} \mathcal{C}_n$  la classe combinatoire des ensembles de descentes :

$$\mathcal{C}_0 := \emptyset \quad \text{et} \quad \mathcal{C}_n := \{I \mid I \subset [n-1]\},$$

pour tout  $n \geq 1$  ; et le type des alphabets infinis totalement ordonnés  $\mathbf{Ens}_{>}^\infty$ .

La donnée  $\text{des}$  qui associe à chaque alphabet infini totalement ordonné  $\mathcal{A}$  :

$$\text{des}_{\mathcal{A}} : \begin{cases} \mathcal{A}^* & \longrightarrow \mathcal{C} \\ w := a_1 a_2 \cdots a_k & \longmapsto \{i \in [k-1] \mid a_i > a_{i+1}\} \end{cases}$$

est une  $\mathbf{Ens}_{>}^\infty$ -application partitionnante. (La preuve est très similaire à celle de l'exemple précédent.)

$$\text{des}_{\mathcal{A}}(\text{ouragan}) = \{2, 3, 5\}.$$

La donnée **val** qui associe à chaque alphabet  $\mathcal{A}$  :

$$\text{val}_{\mathcal{A}} : \begin{cases} \mathcal{A}^* & \longrightarrow \mathcal{C} \\ w := a_1 a_2 \cdots a_k & \longmapsto \begin{cases} \{2 \text{ si } a_1 > a_2\} & \text{si } k = 2, \\ \{i \in \{2, \dots, k-1\} \mid a_{i-1} > a_i \leq a_{i+1}\} & \text{sinon.} \end{cases} \end{cases}$$

(où  $\mathcal{C}$  est l'ensemble des « vallées ») est aussi une  $\text{Ens}_{>}^{\infty}$ -application partitionnante.

$$\text{val}_{\mathcal{A}}(\text{cyclone}) = \{3\}.$$

On peut évidemment considérer des constructions analogues pour les statistiques de *montée* et *pic*.

**Remarque 10.2.6 :** Dans la définition de **val**, on gère le cas où les mots sont de longueur 2. Ce cas est important pour la compatibilité au mélange. Si on oublie ce cas alors on a le problème suivant :

$$\text{val}_{\mathbb{N}}(\textcolor{red}{1}\textcolor{blue}{2}) = \text{val}_{\mathbb{N}}(\textcolor{blue}{2}\textcolor{red}{1}) = \emptyset \quad \text{et} \quad \text{val}_{\mathbb{N} \oplus \mathbb{N}}(\textcolor{blue}{1}\textcolor{red}{1}\textcolor{blue}{2}) = \{2\} \neq \emptyset = \text{val}_{\mathbb{N} \oplus \mathbb{N}}(\textcolor{blue}{1}\textcolor{red}{2}\textcolor{blue}{1}).$$

**Remarque 10.2.7 :** On peut associer une application partitionnante aux autres statistiques : *double montée* et *double descente*. Comme pour les vallées et les pics, on doit traiter à part le cas des mots de longueur 2 pour satisfaire la Condition 5 de compatibilité au mélange.

### 10.2.1. Opérations

Dans cette sous-section, on montre que les applications partitionnantes sont stables pour un certain nombre d'opérations. En particulier, en considérant les relations d'équivalences sous-jacentes aux applications, on observe que l'union et l'intersection de telles relations définissent une relation qui préserve les propriétés des applications partitionnantes.

Ces opérations n'apparaissent pas ou peu dans la littérature. On va donc pouvoir mettre en avant de nouvelles algèbres de Hopf combinatoires.

Soit **Alph** un type d'alphabet et soit  $\phi$  une **Alph**-application partitionnante sur la classe combinatoire  $\mathcal{C}$ .

On note  $\sim_{\phi_{\mathcal{A}}}$  la relation d'équivalence (homogène) de l'application  $\phi_{\mathcal{A}}$  :

$$u \sim_{\phi_{\mathcal{A}}} v \quad \text{si et seulement si} \quad \phi_{\mathcal{A}}(u) = \phi_{\mathcal{A}}(v).$$

En l'absence d'ambiguïté, on note simplement  $\sim_{\phi}$  au lieu  $\sim_{\phi_{\mathcal{A}}}$ .

**Proposition 10.2.8 :** *Le quotient  $\mathcal{A}^* / \sim_{\phi}$  est en bijection avec  $\mathcal{C}$ .*

PREUVE : Découle immédiatement de la Condition 1 de surjectivité de chaque  $\phi_{\mathcal{A}}$ . ■

Réciproquement, à toute relation d'équivalence  $\sim$  sur  $\mathcal{A}^*$ , on peut associer la projection canonique  $\phi_{\sim} : \mathcal{A}^* \rightarrow \mathcal{A}^* / \sim$  qui envoie chaque mot sur sa classe d'équivalence. Cela nous permet de montrer que les opérations naturelles d'union et d'intersection sur les relations d'équivalences associées à des applications partitionnantes préservent la structure d'application partitionnante.

## Union

Soient  $\phi$  et  $\psi$  deux **Alph**-applications partitionnantes.

Soit  $\phi \vee \psi$  la donnée qui associe à chaque alphabet  $\mathcal{A}$  la projection canonique :

$$(\phi \vee \psi)_{\mathcal{A}} : \mathcal{A}^* \longrightarrow \mathcal{A}^*/(\sim_{\phi} \vee \sim_{\psi})$$

où  $(\sim_{\phi} \vee \sim_{\psi})$  est la clôture transitive de l'union des relations d'équivalences (§ 1.1.7) des applications  $\phi_{\mathcal{A}}$  et  $\psi_{\mathcal{A}}$ .

**Proposition 10.2.9 :**  $(\phi \vee \psi)$  est une **Alph**-application partitionnante.

PREUVE: La Proposition 10.2.8 nous assure que les relations  $\sim_{\phi}$  et  $\sim_{\psi}$  sont des relations d'équivalences homogènes. La relation  $\sim_{\phi} \vee \sim_{\psi}$  est donc aussi homogène. On déduit donc que  $\mathcal{A}^*/(\sim_{\phi} \vee \sim_{\psi})$  est (en bijection avec) une classe combinatoire (Définition 1.1.1). On a alors que, quelque soit l'alphabet  $\mathcal{A}$ , l'application  $(\phi \vee \psi)_{\mathcal{A}}$  est surjective (Condition 1) et graduée (Condition 2). Par définition de l'union, on déduit immédiatement que l'application commute avec les injections gauche et droite (Condition 3), qu'elle est compatible aux facteurs (Condition 4) et aussi au mélange (Condition 5). ■

**Exemple 10.2.10 :** L'union  $\text{val} \vee \text{pic}$  (avec  $\text{pic}$  la  $\text{Ens}_{\leq}^{\infty}$ -application partitionnante associée aux pics) est une  $\text{Ens}_{\leq}^{\infty}$ -application partitionnante. Cet exemple n'est pas particulièrement intéressant, car pour tout  $n$ , il n'y a qu'une classe d'équivalence.

## Intersection

On pose  $\phi \wedge \psi$  la donnée qui associe à chaque alphabet  $\mathcal{A}$  la projection canonique :

$$(\phi \wedge \psi)_{\mathcal{A}} : \mathcal{A}^* \longrightarrow \mathcal{A}^*/(\sim_{\phi} \wedge \sim_{\psi})$$

où  $(\sim_{\phi} \wedge \sim_{\psi})$  est l'intersection des relations d'équivalences (§ 1.1.7) des applications  $\phi_{\mathcal{A}}$  et  $\psi_{\mathcal{A}}$ .

**Proposition 10.2.11 :**  $(\phi \wedge \psi)$  est une **Alph**-application partitionnante.

PREUVE: Immédiat. ■

**Exemple 10.2.12 :** L'intersection  $\text{val} \wedge \text{pic}$  est la donnée qui, à chaque alphabet  $\mathcal{A}$ , associe

$$(\text{val} \wedge \text{pic})_{\mathcal{A}} : w \longmapsto (\text{val}_{\mathcal{A}}(w), \text{pic}_{\mathcal{A}}(w)).$$

La suite des nombres de classes d'équivalences est 1, 1, 3, 7, 15, 63, 127, ... soit  $2^n - 1$  pour tout  $n \geq 1$  et 1 si  $n = 0$  [A000225].

```

sage: # Nombre de classes d'équivalence pour *pic*
sage: [len(Pics(Permutations(n)).quotient()) for n in range(8)]
[1, 1, 2, 2, 3, 5, 8, 13]
sage: # Nombre de classes d'équivalence pour *val*
sage: [len(Vallees(Permutations(n)).quotient())
        for n in range(8)]
[1, 1, 2, 2, 3, 5, 8, 13]
sage: # intersection *pic* et *val*
sage: [len(Pics(Permutations(n)).intersection(
        Vallees(Permutations(n))).quotient())
        for n in range(9)]
[1, 1, 1, 3, 7, 15, 31, 63, 127]

```

En plus des opérations sur les relations d'équivalences, on peut appliquer d'autres opérations aux applications partitionnantes :

### Coloration

Soit  $K$  un ensemble quelconque et soient **Alph** un type d'alphabet et  $\mathbf{Alph}^K$  la coloration de **Alph** par l'ensemble  $K$ . Pour chaque **Alph**-application partitionnante  $\phi$ , on construit  $\phi_K$  la donnée qui associe, à chaque alphabet coloré  $\mathcal{A}$  de  $\mathbf{Alph}^K$ , l'application :

$$\phi_{K,\mathcal{A}} : \left\{ \begin{array}{ccc} \mathcal{A}^* & \longrightarrow & \mathcal{C} \bar{\times} K^* \\ (a_1, c_1)(a_2, c_2) \cdots (a_m, c_m) & \longmapsto & (\phi_{\mathcal{A}}(a_1 a_2 \cdots a_m), c_1 c_2 \cdots c_m) \end{array} \right. ,$$

où  $\mathcal{C} \bar{\times} K^*$  est le produit d'Hadamard (des ensembles gradués) :

$$\mathcal{C} \bar{\times} K^* := \bigoplus_{n \geq 0} \mathcal{C}_n \times K^n. \quad (10.1)$$

**Proposition 10.2.13 :**  $\phi_K$  est une  $\mathbf{Alph}^K$ -application partitionnante.

PREUVE : La construction dépend de  $\phi$  et la coloration n'interfère pas avec les conditions des applications partitionnantes. ■

**Exemple 10.2.14 :** À partir d'un ensemble fini de couleurs  $K$ , on va pouvoir construire automatiquement différentes algèbres de Hopf colorées, comme **FQSym**<sup>(l)</sup> [NT04b].

## Produit d'Hadamard

Soient  $\phi$  et  $\psi$  respectivement une **Alph** et une **Alph'** applications partitionnantes sur  $\mathcal{C}$  et  $\mathcal{C}'$ .

On construit la donnée  $\phi \times \psi$  associant à chaque alphabet  $\mathcal{A} := \mathcal{B} \times \mathcal{C}$  du type **Alph**  $\times$  **Alph'** :

$$(\phi \times \psi)_{\mathcal{A}} : \begin{cases} \mathcal{A}^* & \longrightarrow \mathcal{C} \bar{\times} \mathcal{C}' \\ (b_1, c_1)(b_2, c_2) \cdots (b_m, c_m) & \longmapsto (\phi_{\mathcal{B}}(b_1 b_2 \cdots b_m), \psi_{\mathcal{C}}(c_1 c_2 \cdots c_m)) \end{cases} ,$$

(où  $\bar{\times}$  est le produit d'Hadamard des ensembles gradués (10.1)).

**Proposition 10.2.15 :**  $\phi \times \psi$  est une (**Alph**  $\times$  **Alph'**)-application partitionnante.

PREUVE : Par définition du produit d'Hadamard (10.1),  $\mathcal{C} \bar{\times} \mathcal{C}'$  est une classe combinatoire où le degré d'un couple  $(\tau, \nu)$  est le degré de l'une des structures :  $|(\tau, \nu)| := |\tau| = |\nu|$ .

Chaque mot  $w$  de  $\mathcal{A}^n$  se décompose en  $n$  bilettes  $(b, c) \in \mathcal{B} \times \mathcal{C}$  quelconques. On déduit alors que  $(\phi \times \psi)_{\mathcal{A}}$  est surjective (Condition 1) et graduée (Condition 2).

Par définition de la somme d'alphabets du produit cartésien de deux types d'alphabets, comme  $\phi_{\mathcal{B}}$  et  $\psi_{\mathcal{C}}$  commutent avec les injections gauche et droite, on déduit que  $(\phi \times \psi)_{\mathcal{A}}$  commute avec les morphismes injectifs (Condition 3) :

$$\begin{aligned} (\phi \times \psi)_{\mathcal{A}}((b_1, c_1) \cdots (b_m, c_m)) &= (\phi_{\mathcal{B}}(b_1 \cdots b_m), \psi_{\mathcal{C}}(c_1 \cdots c_m)) , \\ &= (\phi_{\mathcal{B} \oplus \mathcal{B}'}(g_{\mathcal{B}, \mathcal{B}'}(b_1 \cdots b_m)), \psi_{\mathcal{C} \oplus \mathcal{C}'}(g_{\mathcal{C}, \mathcal{C}'}(c_1 \cdots c_m))) , \\ &= (\phi \times \psi)_{\mathcal{A} \oplus \mathcal{A}'}(g_{\mathcal{A}, \mathcal{A}'}((b_1, c_1) \cdots (b_m, c_m))) , \end{aligned}$$

avec  $g_{\mathcal{A}, \mathcal{A}'} = (g_{\mathcal{B}, \mathcal{B}'}, g_{\mathcal{C}, \mathcal{C}'})$ , pour tout alphabet  $\mathcal{A}' := \mathcal{B}' \oplus \mathcal{C}'$ . Les Conditions 4 de compatibilité aux facteurs et 5 de compatibilité au mélange découlent immédiatement de la définition de  $\phi \times \psi$ . ■

## 10.3. Algèbre de Hopf $\mathcal{H}_{\phi}$

Après avoir défini les *applications partitionnantes* dans la précédente section, nous allons donner une construction automatique d'algèbres de Hopf à partir de ces dernières. On construit une algèbre de Hopf  $\mathcal{H}_{\phi}$  à partir d'une application partitionnante  $\phi$  sur une classe combinatoire  $\mathcal{C}$ . Cette construction définit une base  $(m_{\tau})$ , dite monomiale, indexée par les structures  $\tau$  de  $\mathcal{C}$ . Dans la première sous-section, on décrit le  $\mathbb{k}$ -espace vectoriel  $\mathcal{H}_{\phi}(\mathcal{A})$  comme un sous-espace de l'algèbre libre  $\mathbb{k}\langle \mathcal{A} \rangle$ . Dans une seconde, on montre que  $\mathcal{H}_{\phi}(\mathcal{A})$  est une sous-algèbre de l'algèbre libre. Dans la troisième, on lui associe une structure de coalgèbre. Enfin dans une quatrième, on montre que le produit et le coproduit sont des morphismes l'un pour l'autre et donc définissent une structure de bialgèbre et par conséquent d'algèbre de Hopf.

### 10.3.1. Espace vectoriel

Soit  $\phi$  une **Alph**-application partitionnante sur la classe combinatoire  $\mathcal{C}$ . Soit  $\mathcal{A}$  un alphabet de **Alph**.

Soit  $\mathcal{H}_\phi(\mathcal{A})$  l'espace vectoriel engendré par les sommes des fibres de l'application  $\phi_{\mathcal{A}}$  :

$$\mathcal{H}_\phi(\mathcal{A}) := \langle m_\tau(\mathcal{A}) \mid \tau \in \mathcal{C} \rangle, \quad \text{où} \quad m_\tau(\mathcal{A}) = \sum_{\substack{w \in \mathcal{A}^* \\ \phi_{\mathcal{A}}(w) = \tau}} w. \quad (10.2)$$

**Proposition 10.3.1 :** *Si  $\phi$  est une application partitionnante alors  $\mathcal{C}$  est une classe combinatoire connexe.*

PREUVE : Comme le mot vide  $\epsilon$  est l'unique mot de degré 0, les Conditions 1 de surjectivité et 2 de graduation, nous assure que  $\#\mathcal{C}_0 = 1$ . ■

**Proposition 10.3.2 :**  $\mathcal{H}_\phi(\mathcal{A})$  est un sous-espace vectoriel gradué connexe de l'algèbre libre  $\mathbb{k}\langle \mathcal{A} \rangle$ , qui admet  $(m_\tau(\mathcal{A}))_{\tau \in \mathcal{C}}$  comme base homogène.

PREUVE : L'espace vectoriel  $\mathcal{H}_\phi(\mathcal{A})$  est défini à partir de l'ensemble des fibres de l'application  $\phi_{\mathcal{A}}$ . Comme  $\phi_{\mathcal{A}}$  est surjective (Condition 1) et que chaque  $m_\tau(\mathcal{A})$  est la somme des mots de la fibre  $\phi_{\mathcal{A}}^{-1}(\tau)$ , les  $m_\tau(\mathcal{A})$  sont donc linéairement indépendants.  $\phi_{\mathcal{A}}$  est une application graduée (Condition 2),  $m_\tau(\mathcal{A})$  est donc homogène de degré  $d := |\tau|$ . Par définition des classes combinatoires,  $\mathcal{C}_n$  est finie ; donc la famille  $(m_\tau(\mathcal{A}))_{|\tau|=d}$  est une base de la composante homogène de degré  $d$  de  $\mathcal{H}_\phi(\mathcal{A})$ . La Proposition 10.3.1 assure que la dimension de  $\mathcal{H}_\phi(\mathcal{A})$  en degré 0 est 1 ; donc  $\mathcal{H}_\phi(\mathcal{A})$  est connexe. ■

**Exemple 10.3.3 :** L'espace vectoriel  $\mathcal{H}_{\text{occ}}(\mathbb{N}_{>0})$  est un sous-espace de  $\mathbb{k}\langle \mathbb{N}_{>0} \rangle$  où  $(m_\lambda)_\lambda$  est une base indexée par les partitions d'ensembles (§ 1.1.2).

$$m_{\{13|25|4\}}(\mathbb{N}_{>0}) = \sum_{a \neq b; a \neq c; b \neq c} abacb = 12132 + 21231 + 23213 + 32312 + \dots + 14134 + \dots$$

### 10.3.2. Algèbre

À partir de ce sous-espace de l'algèbre libre  $\mathbb{k}\langle \mathcal{A} \rangle := (\mathbb{k}\mathcal{A}^*, \mu, \iota)$ , on montre que le produit de concaténation  $\mu$  et l'unité  $\iota$  stabilisent l'espace  $\mathcal{H}_\phi(\mathcal{A})$ .

**Lemme 10.3.4 :**  $(\mathcal{H}_\phi(\mathcal{A}), \mu, \iota)$  est une sous-algèbre graduée connexe de l'algèbre libre  $\mathbb{k}\langle \mathcal{A} \rangle$ .

PREUVE : D'après la Proposition 10.3.2, l'espace  $\mathcal{H}_\phi(\mathcal{A})$  est un sous-espace vectoriel gradué connexe de  $\mathbb{k}\langle \mathcal{A} \rangle$ .

Soient  $\tau$  et  $\nu$  deux structures de  $\mathcal{C}$  respectivement de degrés  $i$  et  $j$ . Dans la preuve de la Proposition 10.3.2, on a montré que  $m_\tau(\mathcal{A})$  et  $m_\nu(\mathcal{A})$  sont homogènes respectivement de degré  $i$  et  $j$  (Condition 2). On déduit donc que le produit des séries :

$$\mu. (m_\tau(\mathcal{A}) \otimes m_\nu(\mathcal{A})) = m_\tau(\mathcal{A}) \cdot m_\nu(\mathcal{A}) = \sum_{\substack{u \in \phi_{\mathcal{A}}^{-1}(\tau) \\ v \in \phi_{\mathcal{A}}^{-1}(\nu)}} u \cdot v$$

est homogène (dans l'algèbre libre large) de degré  $i + j$ . On souhaite montrer que ce produit de l'algèbre libre est bien définie dans  $\mathcal{H}_\phi(\mathcal{A})$  (comme une somme de termes  $m_\gamma(\mathcal{A})$  avec  $\gamma \in \mathcal{C}_{i+j}$ ).

Soient  $\tau, \nu$  et  $\gamma$  trois structures combinatoires de  $\mathcal{C}$  respectivement de degré  $i, j$  et  $k$ . On pose :

$$[\tau, \nu : \gamma]_{\mathcal{A}} := \{w \in \phi_{\mathcal{A}}^{-1}(\gamma) \mid \phi_{\mathcal{A}}(w[1, i]) = \tau \text{ et } \phi_{\mathcal{A}}(w[i+1, k]) = \nu\}, \quad (10.3)$$

(où  $w[i, j]$  est le facteur  $w_i w_{i+1} \cdots w_j$  (1.4)).

S'il existe un mot  $w \in [\tau, \nu : \gamma]_{\mathcal{A}}$  alors  $\phi^{-1}(\gamma) = [\tau, \nu : \gamma]_{\mathcal{A}}$ ; en effet

- par définition  $[\tau, \nu : \gamma]_{\mathcal{A}} \subset \phi_{\mathcal{A}}^{-1}(\gamma)$ , et
- la Condition 4 de compatibilité aux facteurs assure que

$$\phi_{\mathcal{A}}(w[1, i]) = \phi_{\mathcal{A}}(w'[1, i]) = \tau \quad \text{et} \quad \phi_{\mathcal{A}}(w[i+1, k]) = \phi_{\mathcal{A}}(w'[i+1, k]) = \nu$$

pour tout mot  $w'$  tel que  $\phi_{\mathcal{A}}(w') = \gamma = \phi_{\mathcal{A}}(w)$ , d'où  $\phi_{\mathcal{A}}^{-1}(\gamma) \subset [\tau, \nu : \gamma]_{\mathcal{A}}$  (s'il existe un tel  $w$ ).

Par ailleurs, par définition de  $[\tau, \nu : \gamma]_{\mathcal{A}}$ , on a

$$\phi_{\mathcal{A}}^{-1}(\gamma) \subset \{w \in \mathcal{A}^* \mid \phi_{\mathcal{A}}(w[1, i]) = \tau \text{ et } \phi_{\mathcal{A}}(w[i+1, k]) = \nu\}.$$

Donc le produit de concaténation  $\mu.$  stabilise  $\mathcal{H}_\phi(\mathcal{A})$ .

Enfin comme  $\mathcal{C}$  est connexe (Proposition 10.3.1), il existe une unique structure  $\tau$  (telle que  $|\tau| = 0$ ), et la réalisation de cette structure est  $m_\tau(\mathcal{A}) = \epsilon$ . On déduit donc que l'application unité  $\iota.$  stabilise aussi l'espace. Donc  $(\mathcal{H}_\phi(\mathcal{A}), \mu., \iota.)$  est une sous-algèbre graduée connexe de l'algèbre libre  $\mathbb{k}\langle \mathcal{A} \rangle$ . ■

**Exemple 10.3.5 :** Soient  $\tau = \{13 \mid 2 \mid 4\}$  et  $\nu = \{23 \mid 1\}$  deux partitions d'ensembles.

$$\begin{aligned} m_\tau(\mathbb{N}_{>0})m_\nu(\mathbb{N}_{>0}) &= (1213 + 2123 + 2321 + 3132 + \cdots) \cdot (122 + 211 + 133 + 311 + \cdots) \\ &= \underbrace{1213122}_{\{135|267|4\}} + \underbrace{1213211}_{\{1367|25|4\}} + \underbrace{1213133}_{\{135|467|2\}} + \underbrace{1213311}_{\{1367|45|2\}} + \cdots + \underbrace{2123122}_{\{1367|25|4\}} + \\ &\quad \underbrace{2123211}_{\{135|267|4\}} + \underbrace{2123133}_{\{467|13|25\}} + \underbrace{2123311}_{\{267|13|45\}} + \cdots + \underbrace{1213544}_{\{13|67|2|4|5\}} + \cdots \\ &= m_{\{13|67|2|4|5\}}(\mathbb{N}_{>0}) + m_{\{13|45|67|2\}}(\mathbb{N}_{>0}) + m_{\{467|13|2|5\}}(\mathbb{N}_{>0}) + \\ &\quad m_{\{13|25|67|4\}}(\mathbb{N}_{>0}) + m_{\{467|13|25\}}(\mathbb{N}_{>0}) + m_{\{267|13|4|5\}}(\mathbb{N}_{>0}) + \\ &\quad m_{\{267|13|45\}}(\mathbb{N}_{>0}) + m_{\{135|67|2|4\}}(\mathbb{N}_{>0}) + m_{\{135|467|2\}}(\mathbb{N}_{>0}) + \\ &\quad m_{\{135|267|4\}}(\mathbb{N}_{>0}) + m_{\{1367|2|4|5\}}(\mathbb{N}_{>0}) + m_{\{1367|45|2\}}(\mathbb{N}_{>0}) + \\ &\quad m_{\{1367|25|4\}}(\mathbb{N}_{>0}) \end{aligned}$$



**Corollaire 10.3.6 :** Le produit de  $\mathcal{H}_\phi(\mathcal{A})$  dans la base  $(m_\tau(\mathcal{A}))_{\tau \in \mathcal{C}}$  est donnée par : pour tous  $\tau \in \mathcal{C}_i$  et  $\nu \in \mathcal{C}_j$ ,

$$m_\tau(\mathcal{A})m_\nu(\mathcal{A}) = \sum_{\substack{\gamma \in \mathcal{C}_{i+j} \\ [\tau, \nu; \gamma]_{\mathcal{A}} \neq \emptyset}} m_\gamma(\mathcal{A}).$$

**Exemple 10.3.7 :** À partir de la formule du Corollaire, on a :

$$m_{\{13|2\}}(\mathbb{N}_{>0})m_{\{12\}}(\mathbb{N}_{>0}) = m_{\{13|45|2\}}(\mathbb{N}_{>0}) + m_{\{245|13\}}(\mathbb{N}_{>0}) + m_{\{1345|2\}}(\mathbb{N}_{>0})$$

En effet :

$$\begin{aligned} [\{13 | 2\}, \{12\} : \{12345\}] &= \emptyset \\ [\{13 | 2\}, \{12\} : \{2345 | 1\}] &= \emptyset \\ [\{13 | 2\}, \{12\} : \{1345 | 2\},] &= \{12111, 13111, \dots, 21222, 23222, \dots\} \\ [\{13 | 2\}, \{12\} : \{1245 | 3\}] &= \emptyset \\ &\vdots \\ [\{13 | 2\}, \{12\} : \{345 | 12\}] &= \emptyset, \\ [\{13 | 2\}, \{12\} : \{245 | 13\}] &= \{12122, 13133, \dots, 21211, 23233, \dots\}, \\ [\{13 | 2\}, \{12\} : \{235 | 14\}] &= \emptyset, \\ &\vdots \\ [\{13 | 2\}, \{12\} : \{13 | 45 | 2\},] &= \{12133, 12144, \dots, 21233, 21244, \dots\}, \\ &\vdots \end{aligned}$$

Le point essentiel est : « cette construction est indépendante de l'alphabet choisi ».

**Proposition 10.3.8 :** Soient  $\mathcal{A}$  et  $\mathcal{B}$  deux alphabets du type **Alph**. Les algèbres  $\mathcal{H}_\phi(\mathcal{A})$  et  $\mathcal{H}_\phi(\mathcal{B})$  sont isomorphes.

PREUVE : Soit  $f_{\mathcal{A}}^{\mathcal{B}}$  l'application linéaire

$$f_{\mathcal{A}}^{\mathcal{B}} : m_\tau(\mathcal{A}) \longmapsto m_\tau(\mathcal{B}).$$

Les Conditions 1 et 2, respectivement de surjectivité et de graduation, induisent que  $f_{\mathcal{A}}^{\mathcal{B}}$  est un isomorphisme gradué d'espace vectoriel. Il reste à montrer que  $f_{\mathcal{A}}^{\mathcal{B}}$  est un morphisme d'algèbre :

$$f_{\mathcal{A}}^{\mathcal{B}}(m_\tau(\mathcal{A})m_\nu(\mathcal{A})) = f_{\mathcal{A}}^{\mathcal{B}}(m_\tau(\mathcal{A}))f_{\mathcal{A}}^{\mathcal{B}}(m_\nu(\mathcal{A})) = m_\tau(\mathcal{B})m_\nu(\mathcal{B}).$$

On considère la somme d'alphabets  $\mathcal{C} := \mathcal{A} \oplus \mathcal{B}$ . On montre que l'application  $f_{\mathcal{A}}^{\mathcal{C}}$  est un isomorphisme d'algèbre. Soient  $\tau, \nu$  et  $\gamma$  trois structures de  $\mathcal{C}$  respectivement de

degré  $i, j$  et  $k = i + j$ . En reprenant les notations du Lemme 10.3.4, il nous suffit de montrer que pour l'alphabet  $\mathcal{A}$

$$[\tau, \nu : \gamma]_{\mathcal{A}} = \emptyset \quad \text{si et seulement si} \quad [\tau, \nu : \gamma]_{\mathcal{C}} = \emptyset, \quad (10.4)$$

et respectivement pour l'alphabet  $\mathcal{B}$ ,

$$[\tau, \nu : \gamma]_{\mathcal{B}} = \emptyset \quad \text{si et seulement si} \quad [\tau, \nu : \gamma]_{\mathcal{C}} = \emptyset. \quad (10.5)$$

À partir de la Condition 3 de commutativité, si  $w \in [\tau, \nu : \gamma]_{\mathcal{A}}$  alors, par définition de l'extension du morphisme injectif gauche  $g_{\mathcal{A}, \mathcal{B}}$  (cf. Remarque 10.2.1), on a

$$\begin{aligned} \phi_{\mathcal{A}}(w[1, i]) &= \phi_{\mathcal{C}}(g_{\mathcal{A}, \mathcal{B}}(w[1, i])) = \tau, \\ \phi_{\mathcal{A}}(w[i + 1, k]) &= \phi_{\mathcal{C}}(g_{\mathcal{A}, \mathcal{B}}(w[i + 1, k])) = \nu \end{aligned}$$

et

$$\phi_{\mathcal{A}}(w) = \phi_{\mathcal{C}}(g_{\mathcal{A}, \mathcal{B}}(w)) = \gamma.$$

Donc on a bien  $g_{\mathcal{A}, \mathcal{B}}(w) \in [\tau, \nu : \gamma]_{\mathcal{C}}$ . Ceci implique donc

$$\text{si } [\tau, \nu : \gamma]_{\mathcal{A}} \neq \emptyset \quad \text{alors} \quad [\tau, \nu : \gamma]_{\mathcal{C}} \neq \emptyset.$$

Réciproquement, on suppose  $w' \in [\tau, \nu : \gamma]_{\mathcal{C}}$ . À partir de la Condition 1 de surjectivité, il existe nécessairement  $w \in \mathcal{A}^*$  tel que  $\phi_{\mathcal{A}}(w) = \gamma$ . La Condition 3 de commutativité avec les injections vérifie que

$$\phi_{\mathcal{A}}(w) = \phi_{\mathcal{C}}(g_{\mathcal{A}, \mathcal{B}}(w)) = \gamma = \phi_{\mathcal{C}}(w'),$$

et à partir de la Condition 4 de compatibilité aux facteurs (couplée à la Condition 3 de commutativité avec les injections), on déduit que :

$$\phi_{\mathcal{A}}(w[1, i]) = \phi_{\mathcal{C}}(w'[1, i]) = \tau \quad \text{et} \quad \phi_{\mathcal{A}}(w[i + 1, k]) = \phi_{\mathcal{C}}(w'[i + 1, k]) = \nu.$$

D'où

$$\text{si } [\tau, \nu : \gamma]_{\mathcal{C}} \neq \emptyset \quad \text{alors} \quad [\tau, \nu : \gamma]_{\mathcal{A}} \neq \emptyset.$$

Ainsi on déduit (10.4). De la même manière à partir de l'extension de l'injection naturelle droite  $d_{\mathcal{A}, \mathcal{B}}$ , on montre (10.5). On a donc montré que  $f_{\mathcal{A}}^{\mathcal{C}}$  et  $f_{\mathcal{B}}^{\mathcal{C}}$  sont des isomorphismes d'algèbres. Par définition de ces applications, on a

$$f_{\mathcal{A}}^{\mathcal{B}} = f_{\mathcal{A}}^{\mathcal{C}} \circ f_{\mathcal{B}}^{\mathcal{C}-1}.$$

D'où  $f_{\mathcal{A}}^{\mathcal{B}}$  est aussi un isomorphisme gradué d'algèbre. On a donc que les algèbres  $\mathcal{H}_{\phi}(\mathcal{A})$ ,  $\mathcal{H}_{\phi}(\mathcal{B})$  et  $\mathcal{H}_{\phi}(\mathcal{A} \oplus \mathcal{B}) = \mathcal{H}_{\phi}(\mathcal{C})$  sont isomorphes. ■

Ainsi indépendamment de l'alphabet considéré, on a défini une structure d'algèbre  $\mathcal{H}_\phi := (\mathbb{k}\mathcal{C}, \mu, \iota)$  sur les objets combinatoires. Soit  $\mathbb{k}\mathcal{C}$  le  $\mathbb{k}$ -espace vectoriel engendré par les structures de  $\mathcal{C}$  :

$$\mathbb{k}\mathcal{C} := \langle m_\tau \mid \tau \in \mathcal{C} \rangle_{\mathbb{k}}.$$

Soit  $\mathcal{A}$  un alphabet quelconque du type **Alph**. On pose

$$f^{\mathcal{A}} : m_\tau \longmapsto m_\tau(\mathcal{A}).$$

À partir de la Condition 1 de surjectivité, cette application définit un isomorphisme de  $\mathbb{k}$ -espace vectoriel. On définit alors le produit :

$$\mu : \mathbb{k}\mathcal{C} \otimes \mathbb{k}\mathcal{C} \xrightarrow{f^{\mathcal{A}} \otimes f^{\mathcal{A}}} \mathcal{H}_\phi(\mathcal{A}) \otimes \mathcal{H}_\phi(\mathcal{A}) \xrightarrow{\mu} \mathcal{H}_\phi(\mathcal{A}) \xrightarrow{f^{\mathcal{A}^{-1}}} \mathbb{k}\mathcal{C}, \quad (10.6)$$

et l'unité :

$$\iota : \mathbb{k} \xrightarrow{\iota} \mathcal{H}_\phi(\mathcal{A}) \xrightarrow{f^{\mathcal{A}^{-1}}} \mathbb{k}\mathcal{C}.$$

**Lemme 10.3.9 :**  $(\mathbb{k}\mathcal{C}, \mu, \iota)$  est une algèbre graduée connexe.

**Exemple 10.3.10 :** L'application partitionnante **occ** permet de réaliser l'algèbre **WSym** [Wol36 ; RS06] indexée par les partitions d'ensembles.

L'algèbre  $\mathcal{H}_\phi$ , définie à partir de réalisation polynomiale, permet de déduire une règle d'*assemblage* (associative) sur les structures combinatoires de  $\mathcal{C}$ .

### 10.3.3. Coalgèbre

Dans la précédente sous-section, on a réalisé une structure d'algèbre,  $(\mathbb{k}\mathcal{C}, \mu, \iota)$ , à partir d'une application partitionnante  $\phi$ . Dans cette sous-section, on munit le  $\mathbb{k}$ -espace vectoriel  $\mathbb{k}\mathcal{C}$  d'une structure de coalgèbre. La construction qui suit a été introduite dans le folklore dans [Gel+95] par Israel M. GELFAND, Daniel KROB, Alain LASCOUX, Bernard LECLERC, Vladimir S. RETAKH et Jean-Yves THIBON ainsi que dans [DHT02] par Gérard H. E. DUCHAMP, Florent HIVERT et encore J.-Y. THIBON. À partir de l'application partitionnante  $\phi$ , on veut définir le *dédoublément d'alphabets*, noté  $\delta_{\rightleftharpoons}$ , pour toute somme d'alphabet  $\mathcal{A} \oplus \mathcal{B}$ , comme un morphisme de *démélange*

$$\delta_{\mathcal{A} \rightleftharpoons \mathcal{B}} : \mathcal{H}_\phi(\mathcal{A} \oplus \mathcal{B}) \longrightarrow \mathcal{H}_\phi(\mathcal{A}) \otimes \mathcal{H}_\phi(\mathcal{B}).$$

Soit  $\delta_{\rightleftharpoons}$  la donnée qui associe, à toute paire d'alphabets  $(\mathcal{A}, \mathcal{B})$ , l'application définie linéairement sur les mots  $w$  par

$$\delta_{\mathcal{A} \rightleftharpoons \mathcal{B}} : \begin{cases} \mathbb{k}\langle \mathcal{A} \oplus \mathcal{B} \rangle & \longrightarrow & \mathbb{k}\langle \mathcal{A} \rangle \otimes \mathbb{k}\langle \mathcal{B} \rangle \\ w & \longmapsto & w|_{\mathcal{A}} \otimes w|_{\mathcal{B}} \end{cases}, \quad (10.7)$$

où  $w|_{\mathcal{C}}$  est la restriction de  $w$  au plus long sous-mot de  $w$  sur l'alphabet  $\mathcal{C}$  (avec respectivement  $\mathcal{C} = \mathcal{A}$  et  $\mathcal{B}$ ).

**Remarque 10.3.11 :** On verra par la suite (§ 10.3.4) que l'opérateur de démélange est un morphisme d'algèbre.

**Exemple 10.3.12 :** Soit  $\mathbb{N} \oplus \mathbb{N}$  la somme des entiers rouges et des entiers bleus.

$$\delta_{\mathbb{N} \oplus \mathbb{N}}(63622151992933) = 62259293 \otimes 361193.$$

**Proposition 10.3.13 :** Le diagramme suivant commute :

$$\begin{array}{ccc} \mathbb{k}\langle \mathcal{A} \oplus \mathcal{B} \oplus \mathcal{C} \rangle & \xrightarrow{\delta_{\mathcal{A} \oplus (\mathcal{B} \oplus \mathcal{C})}} & \mathbb{k}\langle \mathcal{A} \rangle \otimes \mathbb{k}\langle \mathcal{B} \oplus \mathcal{C} \rangle \\ \delta_{(\mathcal{A} \oplus \mathcal{B}) \oplus \mathcal{C}} \downarrow & & \downarrow \text{Id} \otimes \delta_{\mathcal{B} \oplus \mathcal{C}} \\ \mathbb{k}\langle \mathcal{A} \oplus \mathcal{B} \rangle \otimes \mathbb{k}\langle \mathcal{C} \rangle & \xrightarrow{\delta_{\mathcal{A} \oplus \mathcal{B}} \otimes \text{Id}} & \mathbb{k}\langle \mathcal{A} \rangle \otimes \mathbb{k}\langle \mathcal{B} \rangle \otimes \mathbb{k}\langle \mathcal{C} \rangle \end{array}.$$

PREUVE : Immédiat. ■

**Lemme 10.3.14 :** Soit  $\mathcal{A} \oplus \mathcal{B}$  la somme des alphabets  $\mathcal{A}$  et  $\mathcal{B}$  du type **Alph** et soit  $\phi$  une **Alph**-application partitionnante.

L'image de l'application linéaire  $\delta_{\mathcal{A} \oplus \mathcal{B}}$  restreinte à  $\mathcal{H}_\phi(\mathcal{A} \oplus \mathcal{B})$  est incluse dans l'espace tensoriel  $\mathcal{H}_\phi(\mathcal{A}) \otimes \mathcal{H}_\phi(\mathcal{B})$ .

PREUVE : Soit  $\gamma$  une structure de  $\mathcal{C}$  de degré  $k$ . Par définition, on a

$$\delta_{\mathcal{A} \oplus \mathcal{B}}(m_\gamma(\mathcal{A} \oplus \mathcal{B})) = \sum_{\substack{w \in (\mathcal{A} \oplus \mathcal{B})^* \\ \phi_{\mathcal{A} \oplus \mathcal{B}}(w) = \gamma}} w|_{\mathcal{A}} \otimes w|_{\mathcal{B}}.$$

Comme les sous-mots  $w|_{\mathcal{A}}$  et  $w|_{\mathcal{B}}$  sont complémentaires, on découpe cette somme en fonction de la longueur des sous-mots :

$$= \sum_{i+j=k} \sum_{\substack{\phi_{\mathcal{A} \oplus \mathcal{B}}(w) = \gamma \\ |w|_{\mathcal{A}} = i, |w|_{\mathcal{B}} = j}} w|_{\mathcal{A}} \otimes w|_{\mathcal{B}}.$$

De plus, pour tout mot  $w$ , il existe une unique permutation  $\sigma \in \text{Sh}_{i,j}$  telle que l'égalité suivante soit vérifiée  $(w|_{\mathcal{A}} w|_{\mathcal{B}}) \cdot \sigma = w$ , d'où

$$= \sum_{i+j=k} \sum_{\sigma \in \text{Sh}_{i,j}} \sum_{\substack{u \in \mathcal{A}^i \\ v \in \mathcal{B}^j \\ \phi_{\mathcal{A} \oplus \mathcal{B}}((uv) \cdot \sigma) = \gamma}} u \otimes v.$$

Dans la formule, la somme :

$$\sum_{\substack{u \in \mathcal{A}^i, v \in \mathcal{B}^j \\ \phi_{\mathcal{A} \oplus \mathcal{B}}((uv) \cdot \sigma) = \gamma}} u \otimes v$$

est sans multiplicité. On souhaite montrer que si  $u \otimes v$  apparaît dans cette somme alors le terme  $m_{\tau}(\mathcal{A}) \otimes m_{\nu}(\mathcal{B})$  apparaît totalement (avec  $\tau := \phi_{\mathcal{A}}(u)$  et  $\nu := \phi_{\mathcal{B}}(v)$ ).

Semblablement aux notations du Lemme 10.3.4, on pose

$$[\gamma : (\tau, \nu), \sigma]_{(\mathcal{A}, \mathcal{B})} := \{(u, v) \in \phi_{\mathcal{A}}^{-1}(\tau) \times \phi_{\mathcal{B}}^{-1}(\nu) \mid \phi_{\mathcal{A} \oplus \mathcal{B}}((uv) \cdot \sigma) = \gamma\}$$

l'ensemble des couples  $(u, v)$  qui apparaissent dans le démelange de  $m_{\gamma}(\mathcal{A} \oplus \mathcal{B})$ . À partir de la Condition 5 de mélange, on déduit que s'il existe un tel couple  $(u, v)$  alors

$$[\gamma : (\tau, \nu), \sigma]_{(\mathcal{A}, \mathcal{B})} = \phi_{\mathcal{A}}^{-1}(\tau) \times \phi_{\mathcal{B}}^{-1}(\nu).$$

D'où

$$\sum_{\substack{u \in \mathcal{A}^i, v \in \mathcal{B}^j \\ \phi_{\mathcal{A} \oplus \mathcal{B}}((uv) \cdot \sigma) = \gamma}} u \otimes v = \sum_{\substack{\tau \in \mathcal{C}_i, \nu \in \mathcal{C}_j \\ [\gamma : (\tau, \nu), \sigma]_{(\mathcal{A}, \mathcal{B})} \neq \emptyset}} m_{\tau}(\mathcal{A}) \otimes m_{\nu}(\mathcal{B}).$$

$\delta_{\mathcal{A} \oplus \mathcal{B}}$  définit donc bien une application linéaire :

$$\delta_{\mathcal{A} \oplus \mathcal{B}} : \mathcal{H}_{\phi}(\mathcal{A} \oplus \mathcal{B}) \longrightarrow \mathcal{H}_{\phi}(\mathcal{A}) \otimes \mathcal{H}_{\phi}(\mathcal{B}). \quad \blacksquare$$

**Exemple 10.3.15 :**

$$\begin{aligned} \delta_{\mathbb{N} \oplus \mathbb{N}}(m_{\{13|2\}}(\mathbb{N} \otimes \mathbb{N})) &= \delta_{\mathbb{N} \oplus \mathbb{N}}(\textcolor{red}{121} + \textcolor{red}{131} + \textcolor{red}{232} + \cdots + \textcolor{red}{111} + \textcolor{red}{212} + \textcolor{red}{121} + \cdots \\ &\quad + \textcolor{blue}{111} + \textcolor{blue}{212} + \textcolor{blue}{121} + \cdots + \textcolor{blue}{121} + \textcolor{blue}{131} + \textcolor{blue}{232} + \cdots) \\ &= \textcolor{red}{121} \otimes \epsilon + \textcolor{red}{131} \otimes \epsilon + \textcolor{red}{232} \otimes \epsilon + \cdots + \textcolor{red}{11} \otimes 1 + \textcolor{red}{22} \otimes 1 + \textcolor{red}{11} \otimes 2 + \cdots \\ &\quad + 1 \otimes \textcolor{blue}{11} + 1 \otimes \textcolor{blue}{22} + 2 \otimes \textcolor{blue}{11} + \cdots + \epsilon \otimes \textcolor{blue}{121} + \epsilon \otimes \textcolor{blue}{131} + \epsilon \otimes \textcolor{blue}{232} + \cdots \\ &= m_{\{13|2\}}(\textcolor{red}{\mathbb{N}}) \otimes m_{\emptyset}(\textcolor{blue}{\mathbb{N}}) + m_{\{12\}}(\textcolor{red}{\mathbb{N}}) \otimes m_{\{1\}}(\textcolor{blue}{\mathbb{N}}) + m_{\{1\}}(\textcolor{red}{\mathbb{N}}) \otimes m_{\{12\}}(\textcolor{blue}{\mathbb{N}}) + \\ &\quad m_{\emptyset}(\textcolor{red}{\mathbb{N}}) \otimes m_{\{13|2\}}(\textcolor{blue}{\mathbb{N}}) \end{aligned}$$

On déduit du Lemme la formule de démelange :

**Corollaire 10.3.16 :**

$$\delta_{\mathcal{A} \oplus \mathcal{B}}(m_{\gamma}(\mathcal{A} \oplus \mathcal{B})) = \sum_{i+j=k} \sum_{\tau \in \mathcal{C}_i, \nu \in \mathcal{C}_j} C_{\tau, \nu}^{\gamma} m_{\tau}(\mathcal{A}) \otimes m_{\nu}(\mathcal{B}),$$

où  $C_{\tau, \nu}^{\gamma}$  est le nombre de permutations  $\sigma$  telle que  $[\gamma : (\tau, \nu), \sigma]_{(\mathcal{A} \oplus \mathcal{B})} \neq \emptyset$ , pour toute structure  $\gamma$  de  $\mathcal{C}_k$ .

**Exemple 10.3.17 :** À partir de la formule du Corollaire, on a :

$$\begin{aligned} \delta_{\mathbb{N} \oplus \mathbb{N}}(m_{\{12|3|4\}}(\mathbb{N} \oplus \mathbb{N})) &= m_{\emptyset}(\textcolor{red}{\mathbb{N}}) \otimes m_{\{12|3|4\}}(\textcolor{blue}{\mathbb{N}}) + 2 m_{\{1\}}(\textcolor{red}{\mathbb{N}}) \otimes m_{\{12|3\}}(\textcolor{blue}{\mathbb{N}}) + \\ &\quad m_{\{1|2\}}(\textcolor{red}{\mathbb{N}}) \otimes m_{\{12\}}(\textcolor{blue}{\mathbb{N}}) + m_{\{12\}}(\textcolor{red}{\mathbb{N}}) \otimes m_{\{1|2\}}(\textcolor{blue}{\mathbb{N}}) + \\ &\quad 2 m_{\{12|3\}}(\textcolor{red}{\mathbb{N}}) \otimes m_{\{1\}}(\textcolor{blue}{\mathbb{N}}) + m_{\{13|2|4\}}(\textcolor{red}{\mathbb{N}}) \otimes m_{\emptyset}(\textcolor{blue}{\mathbb{N}}) \end{aligned}$$

Par exemple, le coefficient  $C_{\{12|3\}, \{1\}}^{\{12|3|4\}} = 2$  s'obtient à partir des permutations 1234 et 1243 sur les couples de mots  $(u, v)$  de la forme  $u = \textcolor{red}{aab}$  et  $v = \textcolor{blue}{c}$  :

$$[\{12 \mid 3 \mid 4\} : (\{12 \mid 3\}, \{1\}), 1234]_{(\mathbb{N}, \mathbb{N})} = [\{12 \mid 3 \mid 4\} : (\{12 \mid 3\}, \{1\}), 1243]_{(\mathbb{N}, \mathbb{N})}.$$

En reprenant l'isomorphisme d'espace vectoriel  $f_{\mathcal{A}}^{\mathcal{B}} : m_{\tau}(\mathcal{A}) \mapsto m_{\tau}(\mathcal{B})$ , on établit, encore une fois, une propriété d'« indépendance » du *démélange* vis-à-vis de l'alphabet :

**Proposition 10.3.18 :** Soient  $\gamma, \tau$  et  $\nu$  trois structures de  $\mathcal{C}$ , soit  $\sigma$  une permutation et soient  $\mathcal{A} \oplus \mathcal{B}$  et  $\mathcal{C} \oplus \mathcal{D}$  deux sommes d'alphabets du type **Alph**. On a l'équivalence suivante :

$$[\gamma : (\tau, \nu), \sigma]_{(\mathcal{A}, \mathcal{B})} = \emptyset \quad \text{si et seulement si} \quad [\gamma : (\tau, \nu), \sigma]_{(\mathcal{C}, \mathcal{D})} = \emptyset.$$

PREUVE : On montre que

$$[\gamma : (\tau, \nu), \sigma]_{(\mathcal{A}, \mathcal{B})} = \emptyset \quad \text{si et seulement si} \quad [\gamma : (\tau, \nu), \sigma]_{(\mathcal{A} \oplus \mathcal{C}, \mathcal{B} \oplus \mathcal{D})} = \emptyset \quad (10.8)$$

et

$$[\gamma : (\tau, \nu), \sigma]_{(\mathcal{C}, \mathcal{D})} = \emptyset \quad \text{si et seulement si} \quad [\gamma : (\tau, \nu), \sigma]_{(\mathcal{A} \oplus \mathcal{C}, \mathcal{B} \oplus \mathcal{D})} = \emptyset. \quad (10.9)$$

On pose  $E := \mathcal{A} \oplus \mathcal{C}$  et  $F := \mathcal{B} \oplus \mathcal{D}$ .

À partir de la Condition 3 de commutativité avec les injections, on a

$$\phi_{\mathcal{A}}(u) = \tau = \phi_E(g_{\mathcal{A}, \mathcal{C}}(u)), \quad \phi_{\mathcal{B}}(v) = \nu = \phi_F(g_{\mathcal{B}, \mathcal{D}}(v))$$

et

$$\phi_{\mathcal{A} \oplus \mathcal{B}}((g_{\mathcal{A}, \mathcal{B}}(u)d_{\mathcal{A}, \mathcal{B}}(v)) \cdot \sigma) = \gamma = \phi_{E \oplus F}((g_{E, F}(g_{\mathcal{A}, \mathcal{C}}(u))d_{E, F}(g_{\mathcal{B}, \mathcal{D}}(v))) \cdot \sigma)$$

On a donc que si  $[\gamma : (\tau, \nu), \sigma]_{(\mathcal{C}, \mathcal{D})} \neq \emptyset$  alors  $[\gamma : (\tau, \nu), \sigma]_{(E, F)} \neq \emptyset$ .

Réciproquement s'il existe  $(u', v') \in [\gamma : (\tau, \nu), \sigma]_{(E, F)}$ , d'après la Condition 1 de surjectivité, il existe  $(u, v) \in \mathcal{A}^* \times \mathcal{B}^*$  tels que  $\phi_{\mathcal{A}}(u) = \tau$  et  $\phi_{\mathcal{B}}(v) = \nu$ . À partir de la Condition 3 de commutativité avec les injections, on déduit que

$$\phi_{\mathcal{A}}(u) = \phi_E(g_{\mathcal{A}, \mathcal{C}}(u)) = \tau = \phi_E(u') \quad \text{et} \quad \phi_{\mathcal{B}}(v) = \phi_F(g_{\mathcal{B}, \mathcal{D}}(v)) = \nu = \phi_F(v').$$

À partir de la Condition 5 de compatibilité au mélange (couplé à la Condition 3 de commutativité des injections), on déduit que

$$\phi_{E \oplus F}((g_{E, F}(g_{\mathcal{A}, \mathcal{C}}(u))d_{E, F}(g_{\mathcal{B}, \mathcal{D}}(v))) \cdot \sigma) = \gamma = \phi_{E \oplus F}((g_{E, F}(u')d_{E, F}(v')) \cdot \sigma).$$

Donc si  $[\gamma : (\tau, \nu), \sigma]_{(E, F)} \neq \emptyset$  alors  $[\gamma : (\tau, \nu), \sigma]_{(\mathcal{A}, \mathcal{B})} \neq \emptyset$ , et ainsi on obtient (10.8). On applique le même raisonnement (avec l'injection droite) pour montrer la seconde équivalence (10.9) et ainsi déduire la proposition. ■

À partir de ces différentes propriétés sur  $\delta_{\rightleftharpoons}$ , on peut maintenant définir une structure de coalgèbre sur le  $\mathbb{k}$ -espace vectoriel  $\mathbb{k}\mathcal{C}$ . On définit l'application

$$\Delta : \mathbb{k}\mathcal{C} \xrightarrow{f^{\mathcal{A} \oplus \mathcal{B}}} \mathcal{H}_{\phi}(\mathcal{A} \oplus \mathcal{B}) \xrightarrow{\delta_{\mathcal{A} \rightleftharpoons \mathcal{B}}} \mathcal{H}_{\phi}(\mathcal{A}) \otimes \mathcal{H}_{\phi}(\mathcal{B}) \xrightarrow{f^{\mathcal{A}^{-1}} \otimes f^{\mathcal{B}^{-1}}} \mathbb{k}\mathcal{C} \otimes \mathbb{k}\mathcal{C}. \quad (10.10)$$

À partir des Propositions 10.3.13 et 10.3.18, on déduit que le diagramme suivant commute :

$$\begin{array}{ccc}
\mathbb{k}\langle \mathcal{A}' \oplus \mathcal{B}' \oplus \mathcal{C}' \rangle & \xrightarrow{\delta_{\mathcal{A}' \oplus \mathcal{B}' \oplus \mathcal{C}'}} & \mathbb{k}\langle \mathcal{A}' \rangle \otimes \mathbb{k}\langle \mathcal{B}' \oplus \mathcal{C}' \rangle \\
\downarrow \delta_{\mathcal{A}' \oplus \mathcal{B}' \oplus \mathcal{C}'} & \swarrow f^{\mathcal{A}' \oplus \mathcal{B}' \oplus \mathcal{C}'} \simeq & \swarrow f^{\mathcal{A}' \oplus \mathcal{B}' \oplus \mathcal{C}'} \simeq \\
& \mathbb{k}\mathcal{C} & \mathbb{k}\mathcal{C} \otimes \mathbb{k}\mathcal{C} \\
& \downarrow f^{\mathcal{A} \oplus \mathcal{B} \oplus \mathcal{C}} \simeq & \downarrow f^{\mathcal{A} \oplus \mathcal{B} \oplus \mathcal{C}} \simeq \\
& \mathbb{k}\langle \mathcal{A} \oplus \mathcal{B} \oplus \mathcal{C} \rangle & \xrightarrow{\delta_{\mathcal{A} \oplus \mathcal{B} \oplus \mathcal{C}}} \mathbb{k}\langle \mathcal{A} \rangle \otimes \mathbb{k}\langle \mathcal{B} \oplus \mathcal{C} \rangle \\
& \downarrow \delta_{(\mathcal{A} \oplus \mathcal{B}) \oplus \mathcal{C}} & \downarrow \text{Id} \otimes \delta_{\mathcal{B} \oplus \mathcal{C}} \\
& \mathbb{k}\langle \mathcal{A} \oplus \mathcal{B} \rangle \otimes \mathbb{k}\langle \mathcal{C} \rangle & \xrightarrow{\delta_{\mathcal{A} \oplus \mathcal{B}} \otimes \text{Id}} \mathbb{k}\langle \mathcal{A} \rangle \otimes \mathbb{k}\langle \mathcal{B} \rangle \otimes \mathbb{k}\langle \mathcal{C} \rangle \\
& \downarrow f^{\mathcal{A} \oplus \mathcal{B}} \otimes f^{\mathcal{C}} \simeq & \downarrow f^{\mathcal{A}} \otimes f^{\mathcal{B}} \otimes f^{\mathcal{C}} \simeq \\
& \mathbb{k}\mathcal{C} \otimes \mathbb{k}\mathcal{C} & \mathbb{k}\mathcal{C} \otimes \mathbb{k}\mathcal{C} \otimes \mathbb{k}\mathcal{C} \\
& \downarrow f^{\mathcal{A}' \oplus \mathcal{B}'} \otimes f^{\mathcal{C}'} \simeq & \downarrow f^{\mathcal{A}' \oplus \mathcal{B}'} \otimes f^{\mathcal{C}'} \simeq \\
\mathbb{k}\langle \mathcal{A}' \oplus \mathcal{B}' \rangle \otimes \mathbb{k}\langle \mathcal{C}' \rangle & \xrightarrow{\delta_{\mathcal{A}' \oplus \mathcal{B}'} \otimes \text{Id}} & \mathbb{k}\langle \mathcal{A}' \rangle \otimes \mathbb{k}\langle \mathcal{B}' \rangle \otimes \mathbb{k}\langle \mathcal{C}' \rangle \\
& \downarrow \delta_{\mathcal{A}' \oplus \mathcal{B}'} \otimes \text{Id} & \downarrow \text{Id} \otimes \delta_{\mathcal{B}' \oplus \mathcal{C}'} \\
& \mathbb{k}\langle \mathcal{A}' \oplus \mathcal{B}' \oplus \mathcal{C}' \rangle & \mathbb{k}\langle \mathcal{A}' \oplus \mathcal{B}' \oplus \mathcal{C}' \rangle
\end{array}$$

On déduit que  $\Delta$  est un coproduit (elle satisfait l'axiome de co-associativité (9.3)).

On définit l'application :

$$\varepsilon_{\mathcal{A}} : \begin{cases} \mathcal{H}_{\phi}(\mathcal{A}) & \longrightarrow \mathbb{k} \\ m_{\tau}(\mathcal{A}) & \longmapsto \begin{cases} 1 & \text{si } \tau \in \mathcal{C}_0, \\ 0 & \text{sinon;} \end{cases} \end{cases}$$

À partir de la formule du Corollaire 10.3.16 et de l'isomorphisme (d'espace vectoriel)  $f_{\mathcal{A}}^{\mathcal{B}}$  de  $\mathcal{H}_{\phi}(\mathcal{A})$  vers  $\mathcal{H}_{\phi}(\mathcal{B})$ , on déduit que le diagramme suivant commute :

$$\begin{array}{ccccc}
& & \mathcal{H}_{\phi}(\mathcal{A} \oplus \mathcal{B}) & & \\
& \swarrow f_{\mathcal{A} \oplus \mathcal{B}}^{\mathcal{B}} & \downarrow \delta_{\mathcal{A} \oplus \mathcal{B}} & \searrow f_{\mathcal{A} \oplus \mathcal{B}}^{\mathcal{A}} & \\
& \mathcal{H}_{\phi}(\mathcal{B}) & & \mathcal{H}_{\phi}(\mathcal{A}) & \\
& \swarrow \simeq & & \searrow \simeq & \\
\mathbb{k} \otimes \mathcal{H}_{\phi}(\mathcal{B}) & \xleftarrow{\varepsilon_{\mathcal{A}} \otimes \text{Id}} & \mathcal{H}_{\phi}(\mathcal{A}) \otimes \mathcal{H}_{\phi}(\mathcal{B}) & \xrightarrow{\text{Id} \otimes \varepsilon_{\mathcal{B}}} & \mathcal{H}_{\phi}(\mathcal{A}) \otimes \mathbb{k}
\end{array}$$

En effet, on a : soit  $\gamma \in \mathcal{C}_k$ ,

$$\begin{aligned}
(\varepsilon_{\mathcal{A}} \otimes \text{Id}) \circ \delta_{\mathcal{A} \oplus \mathcal{B}}(m_{\gamma}(\mathcal{A} \oplus \mathcal{B})) &= \left( \sum_{\substack{i+j=k \\ i>0}} \sum_{\substack{\sigma \in \text{Sh}_{i,j} \\ \tau \in \mathcal{C}_i, \nu \in \mathcal{C}_j \\ [\gamma: (\tau, \nu), \sigma] \neq \emptyset}} \underbrace{\varepsilon_{\mathcal{A}}(m_{\tau}(\mathcal{A})) \otimes m_{\nu}(\mathcal{B})}_0 \right) \\
&\quad + \underbrace{\varepsilon_{\mathcal{A}}(m_{\tau_0}(\mathcal{A})) \otimes m_{\gamma}(\mathcal{B})}_1, \\
&= 1 \otimes m_{\gamma}(\mathcal{B}), \\
&\simeq m_{\gamma}(\mathcal{B}) = f_{\mathcal{A} \oplus \mathcal{B}}^{\mathcal{B}}(m_{\gamma}(\mathcal{A} \oplus \mathcal{B})).
\end{aligned}$$

où  $\tau_0$  est l'unique structure de  $\mathcal{C}_0$ . On déduit ainsi que l'application :

$$\varepsilon : \mathbb{k}\mathcal{C} \xrightarrow{f^{\mathcal{A}}} \mathcal{H}_{\phi}(\mathcal{A}) \xrightarrow{\varepsilon_{\mathcal{A}}} \mathbb{k}$$

est une counité pour le coproduit  $\Delta$ , elle satisfait l'axiome de counité (9.4).

**Lemme 10.3.19 :**  $(\mathbb{k}\mathcal{C}, \Delta, \varepsilon)$  est une coalgèbre graduée connexe.

**Exemple 10.3.20 :** La coalgèbre obtenue à partir de l'application partitionnante `occ` est la coalgèbre **WSym** [Wol36 ; RS06] (indexée par les partitions d'ensembles).

Cette seconde structure, la structure de coalgèbre, fournit une manière de *désassembler* (co-associative) des objets combinatoires.

### 10.3.4. Bialgèbre et algèbre de Hopf

Indépendamment dans les deux précédentes sous-sections, on a enrichi le  $\mathbb{k}$ -espace vectoriel  $\mathcal{H}_\phi$  d'une structure d'algèbre et d'une structure de coalgèbre. Dans cette sous-section, on montre que le démélange  $\delta_{\mathcal{A} \rightleftharpoons \mathcal{B}}$  est un morphisme d'algèbre pour le produit de concaténation. On montre par la même occasion dans  $\mathcal{H}_\phi$  le produit et le coproduit sont des morphismes l'un pour l'autre (9.7).

Sur les mots, il est facile de remarquer que le *démélange* est un morphisme pour le produit *concaténation* :

$$\begin{array}{ccc} \mathbb{k}(\mathcal{A} \oplus \mathcal{B})^* \otimes \mathbb{k}(\mathcal{A} \oplus \mathcal{B})^* & \xrightarrow{\mu} & \mathbb{k}(\mathcal{A} \oplus \mathcal{B})^* \xrightarrow{\delta_{\mathcal{A} \rightleftharpoons \mathcal{B}}} \mathbb{k}\mathcal{A}^* \otimes \mathbb{k}\mathcal{B}^* \\ \delta_{\mathcal{A} \rightleftharpoons \mathcal{B}} \otimes \delta_{\mathcal{A} \rightleftharpoons \mathcal{B}} \downarrow & & \uparrow \mu \otimes \mu \\ \mathbb{k}\mathcal{A}^* \otimes \mathbb{k}\mathcal{B}^* \otimes \mathbb{k}\mathcal{A}^* \otimes \mathbb{k}\mathcal{B}^* & \xrightarrow{\text{Id} \otimes \beta \otimes \text{Id}} & \mathbb{k}\mathcal{A}^* \otimes \mathbb{k}\mathcal{A}^* \otimes \mathbb{k}\mathcal{B}^* \otimes \mathbb{k}\mathcal{B}^* \end{array}$$

Quelque soit l'alphabet  $\mathcal{A}$ ,  $\mathcal{H}_\phi(\mathcal{A})$  est un sous-espace vectoriel de  $\mathbb{k}\langle \mathcal{A} \rangle$ . À partir des Lemmes 10.3.4 et 10.3.14, on déduit que le diagramme suivant commute :

$$\begin{array}{ccc} \mathcal{H}_\phi(\mathcal{A} \oplus \mathcal{B}) \otimes \mathcal{H}_\phi(\mathcal{A} \oplus \mathcal{B}) & \xrightarrow{\mu} & \mathcal{H}_\phi(\mathcal{A} \oplus \mathcal{B}) \xrightarrow{\delta_{\mathcal{A} \rightleftharpoons \mathcal{B}}} \mathcal{H}_\phi(\mathcal{A}) \otimes \mathcal{H}_\phi(\mathcal{B}) \\ \delta_{\mathcal{A} \rightleftharpoons \mathcal{B}} \otimes \delta_{\mathcal{A} \rightleftharpoons \mathcal{B}} \downarrow & & \uparrow \mu \otimes \mu \\ \mathcal{H}_\phi(\mathcal{A}) \otimes \mathcal{H}_\phi(\mathcal{B}) \otimes \mathcal{H}_\phi(\mathcal{A}) \otimes \mathcal{H}_\phi(\mathcal{B}) & \xrightarrow{\text{Id} \otimes \beta \otimes \text{Id}} & \mathcal{H}_\phi(\mathcal{A}) \otimes \mathcal{H}_\phi(\mathcal{A}) \otimes \mathcal{H}_\phi(\mathcal{B}) \otimes \mathcal{H}_\phi(\mathcal{B}) \end{array}$$

L'indépendance des opérateurs vis-à-vis de l'alphabet (Propositions 10.3.8 et 10.3.18) induit par la même occasion que  $\mu$  et  $\Delta$  sont des morphismes l'un pour l'autre dans  $\mathcal{H}_\phi$  :

$$\begin{array}{ccc} \mathcal{H}_\phi \otimes \mathcal{H}_\phi & \xrightarrow{\mu} & \mathcal{H}_\phi \xrightarrow{\Delta} \mathcal{H}_\phi \otimes \mathcal{H}_\phi \\ \Delta \downarrow & & \uparrow \mu \otimes \mu \\ \mathcal{H}_\phi \otimes \mathcal{H}_\phi \otimes \mathcal{H}_\phi \otimes \mathcal{H}_\phi & \xrightarrow{\text{Id} \otimes \beta \otimes \text{Id}} & \mathcal{H}_\phi \otimes \mathcal{H}_\phi \otimes \mathcal{H}_\phi \otimes \mathcal{H}_\phi \end{array} \quad (10.11)$$

**Théorème 10.3.21 :**  $(\mathbb{k}\mathcal{C}, \mu, \iota, \Delta, \varepsilon)$  est une algèbre de Hopf graduée connexe.



PREUVE : À partir du Lemme 10.3.9,  $(\mathbb{k}\mathcal{C}, \mu, \iota)$  est une d'algèbre graduée connexe ; et à partir du Lemme 10.3.19,  $(\mathbb{k}\mathcal{C}, \Delta, \varepsilon)$  est une coalgèbre graduée connexe.

Comme le diagramme (10.11) commute,  $(\mathbb{k}\mathcal{C}, \mu, \iota, \Delta, \varepsilon)$  est une bialgèbre graduée connexe et donc une algèbre de Hopf. ■

**Exemple 10.3.22 :** À partir de l'application partitionnante  $\text{occ}$ , on a donc reconstruit algèbre de Hopf  $\text{WSym}$  [Wol36 ; RS06].

## 10.4. Algèbre de Hopf $\mathcal{H}_\phi^\vee$

La précédente section présente une première construction d'algèbre de Hopf à partir d'une application partitionnante  $\phi$ . Dans celle-ci, on définit la « construction duale ». On construit une algèbre de Hopf  $\mathcal{H}_\phi^\vee$ , à partir de l'application partitionnante  $\phi$ , qui définit l'algèbre de Hopf duale de  $\mathcal{H}_\phi$  (le dual gradué).

### 10.4.1. Espace vectoriel

Soit  $\phi$  une **Alph**-application partitionnante sur  $\mathcal{C}$ .

On définit l'espace vectoriel  $\mathcal{H}_\phi^\vee(\mathcal{A})$  comme le quotient de  $\mathbb{k}\mathcal{A}^*$  par les fibres de  $\phi_{\mathcal{A}}$  :

$$\mathcal{H}_\phi^\vee(\mathcal{A}) := \mathbb{k}\mathcal{A}^* / \phi_{\mathcal{A}}.$$

**Proposition 10.4.1 :** L'espace  $\mathcal{H}_\phi^\vee(\mathcal{A})$  est un quotient homogène de  $\mathbb{k}\mathcal{A}^*$ , isomorphe à l'espace vectoriel gradué connexe  $\mathbb{k}\mathcal{C}$ .

PREUVE : D'après la Proposition 10.3.1, on sait que  $\mathbb{k}\mathcal{C}$  est un  $\mathbb{k}$ -espace vectoriel gradué connexe. La Condition 2 de graduation nous permet de déduire que le quotient est homogène :

$$\mathbb{k}\mathcal{A}^* / \phi_{\mathcal{A}} := \bigoplus_{n \geq 0} \mathbb{k}\mathcal{A}^n / \phi_{\mathcal{A}}.$$

De la Condition 1 de surjectivité, on a  $\phi_{\mathcal{A}}(\mathcal{A}^*) \simeq \mathcal{C}$ , donc  $\mathbb{k}\mathcal{A}^* / \phi_{\mathcal{A}} = \mathcal{H}_\phi^\vee(\mathcal{A})$  est isomorphe à l'espace gradué connexe  $\mathbb{k}\mathcal{C}$ . ■

On pose  $(s_\tau(\mathcal{A}))_{\tau \in \mathcal{C}}$  la base de  $\mathcal{H}_\phi^\vee(\mathcal{A})$  définie à partir des fibres de  $\phi_{\mathcal{A}}$  :

$$s_\tau(\mathcal{A}) := \phi_{\mathcal{A}}^{-1}(\tau), \quad \text{pour tout } \tau \in \mathcal{C}.$$

**Remarque 10.4.2 :** Soit  $w$  un mot tel que  $\phi_{\mathcal{A}}(w) = \tau$ . L'élément de base  $s_\tau(\mathcal{A})$  est l'analogue linéaire de la classe d'équivalence de  $w$ , notée  $[w]$ . Pour faciliter les définitions, on s'autorise à identifier  $\tau$  et sa fibre  $\phi_{\mathcal{A}}^{-1}(\tau) = [w]$ . Ainsi, on confond  $s_\tau(\mathcal{A})$  et  $s_{[w]}(\mathcal{A})$ .

On pose aussi  $(s_\tau)_{\tau \in \mathcal{C}}$  la base de  $\mathbb{k}\mathcal{C}$  définie par l'isomorphisme d'espace vectoriel :

$$h^{\mathcal{A}} : s_\tau \longmapsto s_\tau(\mathcal{A}).$$

**Proposition 10.4.3 :** Les  $\mathbb{k}$ -espaces vectoriels  $\mathcal{H}_\phi^\vee(\mathcal{A})$  et  $\mathcal{H}_\phi^\vee(\mathcal{B})$  sont isomorphes.

PREUVE : Immédiat (Proposition 10.4.1). ■

On pose l'isomorphisme d'espace vectoriel :

$$h_{\mathcal{A}}^{\mathcal{B}} : s_\tau(\mathcal{A}) \longmapsto s_\tau(\mathcal{B}). \quad (10.12)$$

**Remarque 10.4.4 :**  $h_{\mathcal{A}}^{\mathcal{B}} = h^{\mathcal{B}} \circ h^{\mathcal{A}-1}$ .

**Exemple 10.4.5 :** À partir de l'application partitionnante **occ**, on obtient l'espace vectoriel  $\mathcal{H}_{\text{occ}}^\vee(\mathbb{N})$  indexé par les partitions d'ensembles (isomorphe à  $\mathcal{H}_{\text{occ}}(\mathbb{N})$ ).

Par exemple, on a

$$s_{\{14|26|3|5\}}(\mathbb{N}) = s_{[123142]}(\mathbb{N}) = \phi_{\mathbb{N}}^{-1}(\{14 \mid 26 \mid 3 \mid 5\}) = \{123142, 123152, 123162, \dots\}$$

## 10.4.2. Algèbre quotient

Soit  $\sqcup_{\mathcal{A}, \mathcal{B}}$  l'application linéaire définie à partir du produit de mélange (1.5) :

$$\sqcup_{\mathcal{A}, \mathcal{B}} : \begin{cases} \mathbb{k}\mathcal{A}^* \otimes \mathbb{k}\mathcal{B}^* & \longrightarrow \mathbb{k}(\mathcal{A} \oplus \mathcal{B})^* \\ u \otimes v & \longmapsto g_{\mathcal{A}, \mathcal{B}}(u) \sqcup d_{\mathcal{A}, \mathcal{B}}(v) \end{cases},$$

pour tous mots  $u \in \mathcal{A}^*$  et  $v \in \mathcal{B}^*$ .

**Proposition 10.4.6 :** L'application  $\sqcup_{\mathcal{A}, \mathcal{B}}$  sur  $\mathbb{k}\mathcal{A}^* \otimes \mathbb{k}\mathcal{B}^*$  passe au quotient par  $\phi$ .

PREUVE : Soient les mots  $u, u' \in \mathcal{A}^m$  tels que  $\phi_{\mathcal{A}}(u) = \phi_{\mathcal{A}}(u')$  et les mots  $v, v' \in \mathcal{B}^n$  tels que  $\phi_{\mathcal{B}}(v) = \phi_{\mathcal{B}}(v')$ . En étendant  $\phi_{\mathcal{A} \oplus \mathcal{B}}$  en une application linéaire, on a

$$\begin{aligned} \phi_{\mathcal{A} \oplus \mathcal{B}}(u \sqcup_{\mathcal{A}, \mathcal{B}} v) &= \phi_{\mathcal{A} \oplus \mathcal{B}}(g_{\mathcal{A}, \mathcal{B}}(u) \sqcup d_{\mathcal{A}, \mathcal{B}}(v)), \\ &= \sum_{\sigma \in \text{Sh}_{m, n}} \phi_{\mathcal{A} \oplus \mathcal{B}}((uv) \cdot \sigma), \end{aligned} \quad [\text{Éq. 1.7}]$$

en utilisant la Condition 5 de compatibilité au mélange, on a

$$\begin{aligned} &= \sum_{\sigma \in \text{Sh}_{m, n}} \phi_{\mathcal{A} \oplus \mathcal{B}}((u'v') \cdot \sigma), \\ &= \phi_{\mathcal{A} \oplus \mathcal{B}}(u' \sqcup_{\mathcal{A}, \mathcal{B}} v'). \end{aligned}$$

Donc l'application  $\sqcup_{\mathcal{A}, \mathcal{B}}$  est compatible avec  $\phi$ . ■

On a, pour  $\tau \in \mathcal{C}_m$  et  $\nu \in \mathcal{C}_n$ ,

$$\begin{aligned} s_\tau(\mathcal{A}) \sqcup_{\mathcal{A}, \mathcal{B}} s_\nu(\mathcal{B}) &= s_{[u]}(\mathcal{A}) \sqcup_{\mathcal{A}, \mathcal{B}} s_{[v]}(\mathcal{B}) \\ &= \sum_{\sigma \in \text{Sh}_{m,n}} s_{[(uv) \cdot \sigma]}(\mathcal{A} \oplus \mathcal{B}), \\ &= \sum_{\sigma \in \text{Sh}_{m,n}} s_{\phi_{\mathcal{A} \oplus \mathcal{B}}((uv) \cdot \sigma)}(\mathcal{A} \oplus \mathcal{B}), \end{aligned}$$

avec  $u \in \phi_{\mathcal{A}}^{-1}(\tau)$  et  $v \in \phi_{\mathcal{B}}^{-1}(\nu)$ .

**Exemple 10.4.7 :**

$$\begin{aligned} s_{\{13|2\}}(\mathbb{N}) \sqcup_{\mathbb{N}, \mathbb{N}} s_{\{1|2\}}(\mathbb{N}) &= s_{[121]}(\mathbb{N}) \sqcup_{\mathcal{A}, \mathcal{B}} s_{[12]}(\mathbb{N}) \\ &= \sum_{\sigma \in \text{Sh}_{3,2}} s_{[(12112) \cdot \sigma]}(\mathbb{N} \oplus \mathbb{N}) \\ &= s_{[(12112)]}(\mathbb{N} \oplus \mathbb{N}) + s_{[(12112)]}(\mathbb{N} \oplus \mathbb{N}) + s_{[(12112)]}(\mathbb{N} \oplus \mathbb{N}) + \\ &\quad s_{[(11212)]}(\mathbb{N} \oplus \mathbb{N}) + s_{[(12121)]}(\mathbb{N} \oplus \mathbb{N}) + \dots \\ &= s_{\{35|1|2|4\}}(\mathbb{N} \oplus \mathbb{N}) + s_{\{24|1|3|5\}}(\mathbb{N} \oplus \mathbb{N}) + 2s_{\{25|1|3|4\}}(\mathbb{N} \oplus \mathbb{N}) + \\ &\quad s_{\{13|2|4|5\}}(\mathbb{N} \oplus \mathbb{N}) + 2s_{\{14|2|3|5\}}(\mathbb{N} \oplus \mathbb{N}) + 3s_{\{15|2|3|4\}}(\mathbb{N} \oplus \mathbb{N}) \end{aligned}$$

On montre par ailleurs que le résultat ne dépend pas des alphabets :

**Proposition 10.4.8 :** L'isomorphisme d'espace vectoriel  $h_{\mathcal{A}}^{\mathcal{B}}$  satisfait l'identité suivante :

$$h_{\mathcal{A} \oplus \mathcal{B}}^{\mathcal{A}' \oplus \mathcal{B}'}(s_\tau(\mathcal{A}) \sqcup_{\mathcal{A}, \mathcal{B}} s_\nu(\mathcal{B})) = h_{\mathcal{A}}^{\mathcal{A}'}(s_\tau(\mathcal{A})) \sqcup_{\mathcal{A}', \mathcal{B}'} h_{\mathcal{B}}^{\mathcal{B}'}(s_\nu(\mathcal{B})).$$

PREUVE : Soient  $\mathcal{A}, \mathcal{A}', \mathcal{B}$  et  $\mathcal{B}'$  quatre alphabets. Soient  $u \in \mathcal{A}^*$  et  $u' \in \mathcal{A}'^*$  tels que

$$\phi_{\mathcal{A}}(u) = \phi_{\mathcal{A}'}(u') = \tau,$$

et soient  $v \in \mathcal{B}^*$  et  $v' \in \mathcal{B}'^*$  tels que

$$\phi_{\mathcal{B}}(v) = \phi_{\mathcal{B}'}(v') = \nu,$$

avec  $\tau$  et  $\nu$  respectivement dans  $\mathcal{C}_m$  et  $\mathcal{C}_n$ .

À partir de la Condition 3 de commutativité avec les injections, on a :

$$\phi_{\mathcal{A}}(u) = \phi_{\mathcal{A} \oplus \mathcal{A}'}(g_{\mathcal{A}, \mathcal{A}'}(u)) = \tau = \phi_{\mathcal{A} \oplus \mathcal{A}'}(d_{\mathcal{A}, \mathcal{A}'}(u')) = \phi_{\mathcal{A}'}(u'),$$

et

$$\phi_{\mathcal{B}}(v) = \phi_{\mathcal{B} \oplus \mathcal{B}'}(g_{\mathcal{B}, \mathcal{B}'}(v)) = \nu = \phi_{\mathcal{B} \oplus \mathcal{B}'}(d_{\mathcal{B}, \mathcal{B}'}(v')) = \phi_{\mathcal{B}'}(v').$$

À partir de la Condition 5 de compatibilité au mélange, on a

$$\begin{aligned} \phi_{\mathcal{A} \oplus \mathcal{B}}((uv) \cdot \sigma) &= \phi_{(\mathcal{A} \oplus \mathcal{A}') \oplus (\mathcal{B} \oplus \mathcal{B}')}((uv) \cdot \sigma), \\ &= \phi_{(\mathcal{A} \oplus \mathcal{A}') \oplus (\mathcal{B} \oplus \mathcal{B}')}((u'v') \cdot \sigma) = \phi_{\mathcal{A}' \oplus \mathcal{B}'}((u'v') \cdot \sigma), \end{aligned}$$

pour tout  $\sigma \in \text{Sh}_{m,n}$ .

L'identité est donc satisfaite. ■

On a donc montré que le produit de mélange  $\sqcup_{\mathcal{A}, \mathcal{B}}$  est « compatible avec l'application partitionnante  $\phi$  » et que le résultat ne dépend pas des alphabets considérés. Ces résultats permettent donc de construire un produit sur  $\mathbb{k}\mathcal{C}$  :

$$\mu_{\sqcup} : \mathbb{k}\mathcal{C} \otimes \mathbb{k}\mathcal{C} \xrightarrow{h^{\mathcal{A}} \otimes h^{\mathcal{B}}} \mathcal{H}_{\phi}^{\vee}(\mathcal{A}) \otimes \mathcal{H}_{\phi}^{\vee}(\mathcal{B}) \xrightarrow{\sqcup_{\mathcal{A}, \mathcal{B}}} \mathcal{H}_{\phi}^{\vee}(\mathcal{A} \oplus \mathcal{B}) \xrightarrow{(h^{\mathcal{A} \oplus \mathcal{B}})^{-1}} \mathbb{k}\mathcal{C}$$

**Exemple 10.4.9 :**

$$\mu_{\sqcup}(s_{\{12\}} \otimes s_{\{1|2\}}) = s_{\{12|3|4\}} + s_{\{13|2|4\}} + s_{\{14|2|3\}} + s_{\{23|1|4\}} s_{\{24|1|3\}} + s_{\{34|1|2\}}$$

Le produit de mélange est une opération associative (Proposition 1.1.19), on déduit que les opérateurs  $\sqcup_{\mathcal{A}, \mathcal{B}}$  satisfont l'identité :

$$\sqcup_{\mathcal{A}, \mathcal{B} \oplus \mathcal{C}} \circ (\text{Id} \otimes \sqcup_{\mathcal{B}, \mathcal{C}}) = \sqcup_{\mathcal{A} \oplus \mathcal{B}, \mathcal{C}} \circ (\sqcup_{\mathcal{A}, \mathcal{B}} \otimes \text{Id}),$$

pour tous alphabets  $\mathcal{A}, \mathcal{B}$  et  $\mathcal{C}$ . On déduit donc que  $\mu_{\sqcup}$  est associatif. Par définition du produit de mélange (1.5), le mot vide  $\epsilon$  est l'élément neutre, d'où

$$f_{\mathcal{A} \oplus \mathcal{B}}^{\mathcal{B}}(s_{[\epsilon]}(\mathcal{A}) \sqcup_{\mathcal{A}, \mathcal{B}} s_{\tau}(\mathcal{B})) = s_{\tau}(\mathcal{B}) = f_{\mathcal{B} \oplus \mathcal{A}}^{\mathcal{B}}(s_{\tau}(\mathcal{B})) \sqcup_{\mathcal{A}, \mathcal{B}} s_{[\epsilon]}(\mathcal{A}).$$

Comme  $\mathcal{C}$  est connexe (Proposition 10.3.1), l'application

$$\iota : \begin{cases} \mathbb{k} & \longrightarrow \mathbb{k}\mathcal{C} \\ \alpha & \longmapsto \alpha s_{[\epsilon]} \end{cases},$$

définit l'unité pour le produit  $\mu_{\sqcup}$ .

**Lemme 10.4.10 :** *Le triplet  $(\mathbb{k}\mathcal{C}, \mu_{\sqcup}, \iota)$  est une algèbre graduée connexe.*

### 10.4.3. Coalgèbre

De la même manière, on considère  $\triangleright_{\mathcal{A}} : \mathbb{k}\mathcal{A}^* \rightarrow \mathbb{k}\mathcal{A}^* \otimes \mathbb{k}\mathcal{A}^*$  le coproduit de déconcaténation défini sur les mots :

$$\triangleright_{\mathcal{A}}(w) := \sum_{i=0}^n w[1, i] \otimes w[i+1, n], \quad (10.13)$$

pour tout mot  $w \in \mathcal{A}^n$  (pour tout  $n \geq 0$ ), tout alphabet  $\mathcal{A}$ , et étendue par linéarité. (Cette opération est évidemment co-associative.)

À partir de la Condition 4 de compatibilité aux facteurs, on déduit que le coproduit de déconcaténation passe au quotient par les fibres de  $\phi_{\mathcal{A}}$  :

$$\triangleright_{\mathcal{A}} : \mathcal{H}_{\phi}^{\vee}(\mathcal{A}) \longrightarrow \mathcal{H}_{\phi}^{\vee}(\mathcal{A}) \otimes \mathcal{H}_{\phi}^{\vee}(\mathcal{A}).$$

À partir de la Condition 3 de commutativité avec les injections, on déduit l'identité suivante :

$$\triangleright_{\mathcal{B}} \circ h_{\mathcal{A}}^{\mathcal{B}} = (h_{\mathcal{A}}^{\mathcal{B}} \otimes h_{\mathcal{A}}^{\mathcal{B}}) \circ \triangleright_{\mathcal{A}}.$$

Autrement dit, le coproduit de déconcaténation agit indépendamment de l'alphabet et l'application

$$\Delta^{\cdot} : \mathbb{k}\mathcal{C} \xrightarrow{h^{\mathcal{A}}} \mathcal{H}_{\phi}^{\vee}(\mathcal{A}) \xrightarrow{\triangleright_{\mathcal{A}}} \mathcal{H}_{\phi}^{\vee}(\mathcal{A}) \otimes \mathcal{H}_{\phi}^{\vee}(\mathcal{A}) \xrightarrow{h^{\mathcal{A}-1} \otimes h^{\mathcal{A}-1}} \mathbb{k}\mathcal{C} \otimes \mathbb{k}\mathcal{C},$$

définie donc un coproduit (indépendamment de l'alphabet  $\mathcal{A}$  considéré).

**Exemple 10.4.11 :**

$$\begin{aligned} \Delta^{\cdot}(s_{\{13|25|4\}}) &= s_{\{13|25|4\}} \otimes s_{\emptyset} + s_{\{13|2|4\}} \otimes s_{\{1\}} + s_{\{13|2\}} \otimes s_{\{1|2\}} + \\ &\quad s_{\{1|2\}} \otimes s_{\{1|2|3\}} + s_{\{1\}} \otimes s_{\{14|2|3\}} + s_{\emptyset} \otimes s_{\{13|25|4\}} \end{aligned}$$

Comme dans la précédente section, on définit linéairement par

$$\varepsilon : \begin{cases} \mathbb{k}\mathcal{C} & \longrightarrow \mathbb{k} \\ s_{\tau} & \longmapsto \begin{cases} 1 & \text{si } \tau \in \mathcal{C}_0, \\ 0 & \text{sinon.} \end{cases} \end{cases}$$

l'application counité.

**Lemme 10.4.12 :** *Le triplet  $(\mathbb{k}\mathcal{C}, \Delta^{\cdot}, \varepsilon)$  est une coalgèbre graduée connexe.*

#### 10.4.4. Algèbre de Hopf

À partir des structures d'algèbre et de coalgèbre obtenues, on obtient une algèbre de Hopf combinatoire :

**Théorème 10.4.13 :**  *$(\mathbb{k}\mathcal{C}, \mu_{\sqcup}, \iota, \Delta^{\cdot}, \varepsilon)$  est une algèbre de Hopf graduée connexe.*

PREUVE: Le triplet  $(\mathbb{k}\mathcal{C}, \mu_{\sqcup}, \iota)$  est une algèbre graduée connexe (Lemme 10.4.10), et le triplet  $(\mathbb{k}\mathcal{C}, \Delta^{\cdot}, \varepsilon)$  est une coalgèbre graduée connexe (Lemme 10.4.12).

On montre que l'axiome 9.7 de compatibilité entre le produit et le coproduit est satisfait en montrant que l'identité suivante sur les mots est satisfaite :

$$\triangleright_{\mathcal{A} \oplus \mathcal{B}} \circ \sqcup_{\mathcal{A}, \mathcal{B}} = (\sqcup_{\mathcal{A}, \mathcal{B}} \otimes \sqcup_{\mathcal{A}, \mathcal{B}}) \circ (\text{Id} \otimes \beta \otimes \text{Id}) \circ (\triangleright_{\mathcal{A}} \otimes \triangleright_{\mathcal{B}}). \quad (10.14)$$

Soient  $u \in \mathcal{A}^m$  et  $v \in \mathcal{B}^n$  deux mots. À partir des définitions de déconcaténation et de mélange, on a

$$\triangleright_{\mathcal{A} \oplus \mathcal{B}} \circ \sqcup_{\mathcal{A}, \mathcal{B}}(u \otimes v) = \sum_{\sigma \in \text{Sh}_{m,n}} \sum_{i=0}^{m+n} w_{\sigma}[1, i] \otimes w_{\sigma}[i+1, m+n],$$

où  $w_{\sigma} := (uv) \cdot \sigma$  ; et

$$(\sqcup_{\mathcal{A}, \mathcal{B}} \otimes \sqcup_{\mathcal{A}, \mathcal{B}}) \circ \beta \circ (\triangleright_{\mathcal{A}} \otimes \triangleright_{\mathcal{B}})(u \otimes v) = \sum_{\substack{0 \leq j \leq m \\ 0 \leq k \leq n}} \sum_{\substack{\sigma_1 \in \text{Sh}_{j,k} \\ \sigma_2 \in \text{Sh}_{m-j, n-k}}} (u_{[j]}v_{[k]}) \cdot \sigma_1 \otimes (u_{[j]}v_{[k]}) \cdot \sigma_2 .$$

avec les notations  $u_{[i]} := u[1, i]$  et  $u_{[i]} := u[i+1, |w|]$  (respectivement avec  $v$ ), et où, par abus,  $\beta$  dénote  $\text{Id} \otimes \beta \otimes \text{Id}$ .

Pour  $u$  et  $v$  fixés, la première somme est entièrement définie par l'ensemble

$$E_{u,v} = \text{Sh}_{m,n} \times \{0, \dots, m+n\} \quad (10.15)$$

et la seconde par l'ensemble

$$F_{u,v} = \{(\sigma_1, \sigma_2, j, k) \mid \sigma_1 \in \text{Sh}_{j,k} \text{ et } \sigma_2 \in \text{Sh}_{m-j, n-k} \text{ avec } 0 \leq j \leq m \text{ et } 0 \leq k \leq n\} .$$

Pour montrer l'identité (10.14), il suffit de montrer une bijection entre ces deux ensembles  $E_{u,v}$  et  $F_{u,v}$  :

$$\begin{array}{ccc} (\sigma, i) & \longleftrightarrow & (\sigma_1, \sigma_2, j, k) , \\ & \Updownarrow & \\ (w_{\sigma}[1, i], w_{\sigma}[i+1, m+n]) & \longleftrightarrow & ((u_{[j]}v_{[k]}) \cdot \sigma_1, (u_{[j]}v_{[k]}) \cdot \sigma_2) . \end{array}$$

Soit la fonction,

$$\zeta : \begin{cases} E_{u,v} & \longrightarrow & F_{u,v} \\ (\sigma, i) & \longmapsto & (\sigma_1, \sigma_2, j, k) \end{cases}$$

avec

- $\sigma_1 := \text{std}(\sigma[1, i])$ , l'unique permutation de  $[i]$ , c'est-à-dire  $\sigma_1 \in \mathcal{S}_i$ , qui préserve l'ordre total  $\sigma(1) \cdots \sigma(i)$  ;
- $\sigma_2 := \text{std}(\sigma[i+1, m+n])$ , l'unique permutation de  $[m+n-i]$  qui préserve l'ordre  $\sigma(i+1) \cdots \sigma(m+n)$  ;
- $j := \ell(w_{\sigma}[1, i]_{|\mathcal{A}})$ , la longueur du préfixe  $w_{\sigma}[1, i]$  restreint aux symboles de  $\mathcal{A}$  ;
- $k := \ell(w_{\sigma}[1, i]_{|\mathcal{B}})$ , la longueur du préfixe  $w_{\sigma}[1, i]$  restreint aux symboles de  $\mathcal{B}$ .

Par construction, les permutations  $\sigma_1$  et  $\sigma_2$  sont respectivement des permutations de  $\text{Sh}_{j,k}$  et  $\text{Sh}_{m-j, n-k}$ , et chaque élément  $(\sigma_1, \sigma_2, j, k)$  s'obtient uniquement à partir de  $\sigma$  et  $i$ . En effet, le terme  $w_{\sigma}[1, i] \otimes w_{\sigma}[i+1, m+n]$  n'apparaît qu'une fois dans la somme, et par définition de  $\zeta$ , on a l'identité suivante :

$$w_{\sigma}[1, i] = (u_{[j]}v_{[k]}) \cdot \sigma_1 \quad \text{et} \quad w_{\sigma}[i+1, m+n] = (u_{[j]}v_{[k]}) \cdot \sigma_2 .$$

La fonction  $\zeta$  est donc injective.

Réciproquement,

$$\zeta' : \begin{cases} F_{u,v} & \longrightarrow & E_{u,v} \\ (\sigma_1, \sigma_2, j, k) & \longmapsto & (\sigma, i) \end{cases}$$

avec

- $i := j + k$ ;
- $\sigma := \sigma(1)\sigma(2) \cdots \sigma(m+n)$  avec

$$\sigma(h) = \begin{cases} \sigma_1(h) & \text{si } \sigma_1(h) \leq j \text{ avec } h \leq i, \\ \sigma_1(h) + m - j & \text{si } \sigma_1(h) > j \text{ avec } h \leq i, \\ \sigma_2(h - i) + j & \text{si } \sigma_2(h - i) \leq k \text{ avec } h > i, \\ \sigma_2(h - i) + i & \text{si } \sigma_2(h - i) > k \text{ avec } h > i. \end{cases}$$

Par construction,  $\sigma$  est une permutation de  $\text{Sh}_{m,n}$  et s'obtient de manière unique à partir de  $\sigma_1, \sigma_2, j$  et  $k$ . L'application  $\zeta'$  est donc aussi injective.

Les applications  $\zeta$  et  $\zeta'$  sont inverses l'une de l'autre; elles définissent donc une bijection. L'identité (10.14) est donc satisfaite. On déduit ainsi que  $(\mathbb{K}\mathcal{C}, \mu_{\sqcup}, \iota, \Delta', \varepsilon)$  est une bialgèbre graduée connexe, d'où une algèbre de Hopf. ■

**Exemple 10.4.14 :** À partir de l'application partitionnante  $\text{occ}$ , cette construction définit le dual gradué de l'algèbre de Hopf **WSym**.

#### 10.4.5. Dualité entre $\mathcal{H}_\phi$ et $\mathcal{H}_\phi^\vee$

Dans cette section et la précédente, on a donné deux constructions d'algèbres de Hopf à partir d'une application partitionnante  $\phi$ . Ces constructions définissent deux algèbres de Hopf en dualité :

**Théorème 10.4.15 :** Les algèbres de Hopf  $\mathcal{H}_\phi$  et  $\mathcal{H}_\phi^\vee$  sont en dualité et  $(m_\tau)$  est la base duale de  $(s_\tau)$ .

PREUVE : Soit  $(m_\tau^\vee)$  la base duale de  $(m_\tau)$ . Par dualité, on a

$$\Delta^\vee(m_\gamma^\vee) = \sum_{\tau, \nu} \langle m_\tau m_\nu \mid m_\gamma^\vee \rangle m_\tau^\vee \otimes m_\nu^\vee,$$

avec  $\langle m_\tau \mid m_\nu^\vee \rangle := \delta_{\tau, \nu}$ .

Par définition du produit dans la base  $(m_\tau)$  (Corollaire 10.3.6), on sait que  $m_\gamma$  apparaît dans le produit de  $m_\tau$  et  $m_\nu$  si et seulement si  $[\tau, \nu : \gamma]_{\mathcal{A}} \neq \emptyset$  (pour  $\mathcal{A}$  un alphabet quelconque). D'après la Condition 2 de graduation, on a :

$$\Delta^*(m_\gamma^\vee) = \sum_{i=0}^{k:=|\gamma|} \sum_{\substack{\tau \in \mathcal{C}_i \\ \nu \in \mathcal{C}_{k-i} \\ [\tau, \nu : \gamma]_{\mathcal{A}} \neq \emptyset}} m_\tau^\vee \otimes m_\nu^\vee.$$

À partir de la Condition 4 de compatibilité aux facteurs, pour tout  $i, j$  tels que  $i+j = k$ , il existe un unique couple  $(\tau, \nu)$  dans  $\mathcal{C}_i \times \mathcal{C}_j$  tel que  $[\tau, \nu : \gamma]_{\mathcal{A}} \neq \emptyset$ . On déduit alors, de la formule du coproduit dans la base  $(s_\tau)$  (défini sur les mots à l'Équation 10.13) et

de la définition de  $[\tau, \nu : \gamma]_{\mathcal{A}}$  (10.3), que les bases  $(m_{\tau}^{\vee})$  et  $(s_{\tau})$  sont isomorphes pour le coproduit.

Soit  $(s_{\tau}^{\vee})$  la base duale de  $(s_{\tau})$ . Par dualité, on a :

$$\Delta^{\vee}(s_{\gamma}^{\vee}) = \sum_{\tau, \nu} \langle s_{\tau} s_{\nu} \mid s_{\gamma}^{\vee} \rangle s_{\tau}^{\vee} \otimes s_{\nu}^{\vee}$$

Par définition du produit dans la base  $(s_{\tau})$ , le terme  $s_{\gamma}$  apparaît autant de fois qu'il existe de permutations  $\sigma$  telle que  $[\gamma : (\tau, \nu), \sigma] \neq \emptyset$ . Comme le produit est gradué, on déduit du Corollaire 10.3.16 que les bases  $(s_{\tau}^{\vee})$  et  $(m_{\tau})$  sont isomorphes pour le coproduit.

Les bases  $(m_{\tau})$  et  $(s_{\tau})$  sont donc en dualité et donc les algèbres de Hopf  $\mathcal{H}_{\phi}$  et  $\mathcal{H}_{\phi}^{\vee}$  sont en dualité. ■



# 11. Extensions

Le chapitre précédent formalise la notion d'*applications partitionnantes*. Ces applications, des mots dans les classes combinatoires, permettent de construire des algèbres de Hopf à partir de propriétés combinatoires élémentaires sur les mots.

Dans ce chapitre, on s'intéresse à remarquer les structures et morphismes communs à ces algèbres de Hopf combinatoires définies à partir d'applications partitionnantes.

La plupart des algèbres de Hopf existantes, définies à partir d'une application partitionnante, sont indexées par des structures combinatoires (au sens de la théorie des espèces). Dans de nombreux cas, il existe des morphismes d'algèbres de Hopf, de ces algèbres, indexées par des structures, vers des algèbres de Hopf indexées par les types d'isomorphismes de ces structures. Au niveau des réalisations, ces morphismes correspondent à la surjection canonique de l'algèbre libre  $\mathbb{k}\langle \mathcal{A} \rangle$  vers l'algèbre des polynômes  $\mathbb{k}[\mathcal{A}]$ . Nous étudierons ce cas dans la première section.

Par ailleurs, ces algèbres de Hopf combinatoire  $\mathcal{H}_\phi$  sont généralement enrichies d'autres structures algébriques : produit  $\#$ , structure dendriforme, produit interne, etc.. Dans les deux sections suivantes, on montre, à partir du formalisme des applications partitionnantes, que ces algèbres de Hopf  $\mathcal{H}_\phi$  définies à partir de l'application partitionnante  $\phi$  sont automatiquement munies d'un produit  $\#$  et d'une structure de bialgèbre codendriforme.

## 11.1. Quotient commutatif

Soit  $\mathbf{F}$  une espèce de structures et soit  $\phi$  une **Alph**-applications partitionnantes sur la classe combinatoire  $\mathcal{C}_\mathbf{F}$  des structures biens-étiquetées de  $\mathbf{F}$ .

**Remarque 11.1.1 :** On rappelle que les classes combinatoires  $\mathcal{C}_\mathbf{F}$  et  $\tilde{\mathcal{C}}_\mathbf{F}$  sont définies par

$$\mathcal{C}_\mathbf{F} := \sum_{n \geq 0} \mathbf{F}[n] \quad \text{et} \quad \tilde{\mathcal{C}}_\mathbf{F} := \sum_{n \geq 0} T(\mathbf{F}[n]). \quad (1.15)$$

Dans cette section, on s'intéresse au cas où l'application partitionnante *commute avec les actions de groupe* :

$$\phi_{\mathcal{A}}(w \cdot \sigma) = \mathbf{F}[\sigma](\phi_{\mathcal{A}}(w)), \quad \text{pour tout } w \in \mathcal{A}^* \text{ et toute bijection } \sigma. \quad (11.1)$$

**Exemple 11.1.2 :** C'est le cas de `occ` :

$$\begin{aligned} \text{occ}_{\mathbb{N}}(42315132 \cdot \mathbf{67125384}) &= \text{occ}_{\mathbb{N}}(13425321) = \{18 \mid 23 \mid 47 \mid 5 \mid 6\} \\ &= \text{Par}[\mathbf{67125384}](\{28 \mid 37 \mid 46 \mid 1 \mid 5\}). \end{aligned}$$

Soit  $\pi$  la surjection canonique :

$$\pi : \mathbb{k}\mathcal{C}_{\mathbb{F}} \longrightarrow \mathbb{k}\tilde{\mathcal{C}}_{\mathbb{F}},$$

dans le  $\mathbb{k}$ -espace vectoriel coinvariant/quotient pour l'action du groupe symétrique de  $\mathbb{k}\mathcal{C}_{\mathbb{F}}$ .

**Théorème 11.1.3 :** Si  $\phi$  commute avec les actions de groupes alors l'algèbre de Hopf  $\mathcal{H}_{\phi}$  passe au quotient par la projection  $\pi$ .

PREUVE : On reprend la notation  $\theta_{\mathcal{A}}$  pour la surjection canonique :

$$\theta_{\mathcal{A}} : \mathbb{k}\langle \mathcal{A} \rangle \longrightarrow \mathbb{k}[\mathcal{A}]$$

de l'algèbre libre  $\mathbb{k}\langle \mathcal{A} \rangle$  dans l'algèbre des polynômes  $\mathbb{k}[\mathcal{A}]$  en  $\mathcal{A}$ .

Comme  $\mathcal{H}_{\phi}(\mathcal{A})$  est une sous-algèbre de  $\mathbb{k}\langle \mathcal{A} \rangle$ , on déduit immédiatement que la structure d'algèbre passe au quotient suivant la congruence associée à  $\theta_{\mathcal{A}}$ .

On suppose  $\mathcal{A} := \mathcal{B} \oplus \mathcal{C}$ . L'opération  $\delta_{\mathcal{B} \oplus \mathcal{C}}$  se définit sur les mots et est bien définie comme application de  $\mathcal{H}_{\phi}(\mathcal{B} \oplus \mathcal{C})$  dans  $\mathcal{H}_{\phi}(\mathcal{B}) \otimes \mathcal{H}_{\phi}(\mathcal{C})$ . Pour les mêmes raisons que précédemment, l'opérateur  $\delta_{\mathcal{B} \oplus \mathcal{C}}$  passe donc au quotient.

Les Propositions 10.3.8 et 10.3.18 d'indépendance vis-à-vis de l'alphabet considéré permettent de déduire de l'hypothèse (11.1) que le produit  $\mu$  (10.6) et le coproduit  $\Delta$  (10.10) passent au quotient suivant la projection  $\pi$ . ■

**Corollaire 11.1.4 :** L'espace coinvariant de  $\mathcal{H}_{\phi}$  pour l'action du groupe symétrique est indexée par  $\tilde{\mathcal{C}}_{\mathbb{F}}$ .

**Exemple 11.1.5 :** À partir du Théorème 11.1.3 et du Corollaire 11.1.4, on obtient les fonctions symétriques, *Sym*, comme le quotient par l'action du groupe symétrique sur les partitions d'ensembles dans **WSym**.

En appliquant cette technique, on obtient par exemple **QSym** comme quotient de **WQSym** ou encore **CQSym** comme quotient de **PQSym**.

## 11.2. Réalisation de #-algèbre

Dans cette section, on utilise notre formalisme pour montrer que toute algèbre de Hopf définie à partir d'une application partitionnante est munie d'un produit  $\#$ . Ce produit a été introduit par Jean-Christophe AVAL et Xavier VIENNOT dans [AV10] et a été étudié sur quelques algèbres de Hopf combinatoires par Jean-Christophe AVAL,

Jean-Christophe NOVELLI et Jean-Yves THIBON dans [ANT11]. On peut aussi citer le travail de Frédéric CHAPOTON qui s'est intéressé à ce produit du point de vue opéradique [Cha09].

Soit  $\mathcal{A}$  un alphabet. Le *produit dièse*, noté  $\#$ , est défini linéairement sur les mots :

$$ua \# bv := \begin{cases} uav & \text{si } a = b, \\ 0 & \text{sinon;} \end{cases}$$

pour tous mots  $u, v \in \mathcal{A}^*$  et tous symboles  $a, b \in \mathcal{A}$ .

Ce produit vérifie la compatibilité suivante avec le produit de concaténation  $\cdot$  :

$$(u \cdot v) \# w = u \cdot (v \# w) \quad \text{et} \quad (u \# v) \cdot w = u \# (v \cdot w).$$

Soit  $\phi$  une **Alph**-application partitionnante sur  $\mathcal{C}$ .

**Lemme 11.2.1 :** *Le produit dièse stabilise l'espace  $\mathcal{H}_\phi(\mathcal{A})$ .*

On montre ce lemme de la même manière que dans le Lemme 10.3.4 :

PREUVE : Soient  $\tau, \nu$  et  $\gamma$  trois structures de  $\mathcal{C}$ . On pose

$$[\tau \# \nu : \gamma] = \{w \in \phi^{-1}(\gamma) \mid w = uav \text{ tel que } \phi_{\mathcal{A}}(ua) = \tau \text{ et } \phi_{\mathcal{A}}(av) = \nu\}$$

avec  $u, v \in \mathcal{A}^*$  et  $a \in \mathcal{A}$ . Par définition,  $[\tau \# \nu : \gamma] \subset \phi_{\mathcal{A}}^{-1}(\gamma)$ , et à partir de la Condition 4 de compatibilité aux facteurs, on déduit que  $\phi_{\mathcal{A}}^{-1}(\gamma) \subset [\tau \# \nu : \gamma]$ . On obtient donc que  $\phi_{\mathcal{A}}^{-1}(\gamma) = [\tau \# \nu : \gamma]$  (si  $[\tau \# \nu : \gamma] \neq \emptyset$ ) et ainsi que le produit  $\#$  stabilise l'espace  $\mathcal{H}_\phi(\mathcal{A})$ . ■

En utilisant le même raisonnement que la Proposition 10.3.8 avec la Condition 3 de commutativité avec les injections, on montre que l'isomorphisme d'espace vectoriel  $f_{\mathcal{A}}^{\mathcal{B}}$  est un morphisme pour  $\#$  et on a alors

**Théorème 11.2.2 :** *L'algèbre de Hopf  $\mathcal{H}_\phi$  est muni d'un produit dièse :*

$$\mu_{\#} : \mathbb{k}\mathcal{C} \otimes \mathbb{k}\mathcal{C} \xrightarrow{f_{\mathcal{A}} \otimes f_{\mathcal{A}}} \mathcal{H}_\phi(\mathcal{A}) \otimes \mathcal{H}_\phi(\mathcal{A}) \xrightarrow{\#} \mathcal{H}_\phi(\mathcal{A}) \xrightarrow{f_{\mathcal{A}}^{-1}} \mathbb{k}\mathcal{C}.$$

**Corollaire 11.2.3 :** *Le produit dièse est homogène :*

$$\mu_{\#} := \bigoplus_{m, n \geq 1} \mu_{\#, m, n} \quad \text{avec} \quad \mu_{\#, m, n} : \mathbb{k}\mathcal{C}_m \otimes \mathbb{k}\mathcal{C}_n \longrightarrow \mathbb{k}\mathcal{C}_{m+n-1},$$

et si  $\mathcal{C}_1$  est un singleton  $\{\tau\}$  alors  $m_{\tau}$  est l'unité pour  $\#$ .

**Exemple 11.2.4 :** Dans **WSym**, le produit  $\#$  se décrit par : soient  $\tau \in \mathcal{C}_{\text{Par}, i}$  et  $\nu \in \mathcal{C}_{\text{Par}, j}$

$$\mu_{\#}(m_{\tau} \otimes m_{\nu}) = \sum_{\pi \in \tau \# \nu} m_{\nu},$$

où  $\tau \# \nu$  est l'ensemble des partitions  $\pi = \{S_h\}_h$  de l'ensemble  $[i + j - 1]$  telles que :

$$\begin{aligned} \tau &= \{S'_h\}_h \setminus \{\emptyset\}, & \text{avec } S'_h &:= \{k \mid k \in S_h \text{ et } k \leq i\}, \text{ et} \\ \nu &= \{S''_h\}_h \setminus \{\emptyset\}, & \text{avec } S''_h &:= \{k - i + 1 \mid k \in S_h \text{ et } k \geq i\}. \end{aligned}$$

## 11.3. Réalisation dendriforme

Cette section se consacre aux liens entre algèbres de Hopf définies à partir d'applications partitionnantes et les structures dendriformes. Ces dernières ont été introduites par Jean-Louis LODAY [Lod01, § 5] et largement étudiées par Loïc FOISSY avec sa version *bidendriforme* du théorème de Milnor–Moore [Foi07].

Les algèbres de Hopf **FQSym**, **WQSym**, **PQSym**, **PBT**, **WSym** ainsi que de nombreuses autres sont munies de structures dendriformes ou bidendriformes. Ces algèbres de Hopf peuvent généralement être définies à partir d'applications partitionnantes. Nous montrons que de telles algèbres de Hopf sont automatiquement munies d'une structure de bialgèbre codendriforme. Nous étudions aussi le cas des alphabets totalement ordonnés qui fournissent quelques conditions sur les applications partitionnantes telles que l'algèbre de Hopf associée soit munie d'une structure de bialgèbre bidendriforme.

### 11.3.1. Définitions

**Définition 11.3.1** ([Lod01, Définition 5.1]) : Une *algèbre dendriforme*  $E$  est un  $\mathbb{k}$ -espace vectoriel  $E$  muni de deux opérations binaires :

$$\begin{aligned} \prec : E \otimes E &\longrightarrow E, \\ \succ : E \otimes E &\longrightarrow E, \end{aligned}$$

qui satisfont les axiomes :

$$(a \prec b) \prec c = a \prec (b \prec c) + a \prec (b \succ c), \quad (11.2)$$

$$(a \succ b) \prec c = a \succ (b \prec c), \quad (11.3)$$

$$(a \prec b) \succ c + (a \succ b) \succ c = a \succ (b \succ c), \quad (11.4)$$

pour tous  $a, b$  et  $c$  dans  $E$ .

On appelle *produit gauche* et *produit droit* respectivement  $\prec$  et  $\succ$ .

**Proposition 11.3.2** ([Lod01, Lemme 5.2]) : Pour toute algèbre dendriforme  $E$  le produit  $\star$  défini par

$$x \star y = x \prec y + x \succ y,$$

est associatif.

On traduit les axiomes (11.2), (11.3) et (11.4) en termes de diagrammes afin de facilement introduire la notation de *coalgèbre dendriforme*. Les trois axiomes font respectivement commuter les diagrammes suivants :

$$\begin{array}{ccc} E \otimes E \otimes E & \xrightarrow{\prec \otimes \text{Id}} & E \otimes E \\ \text{Id} \otimes \star \downarrow & & \downarrow \prec \\ E \otimes E & \xrightarrow{\prec} & E \end{array}, \quad \begin{array}{ccc} E \otimes E \otimes E & \xrightarrow{\succ \otimes \text{Id}} & E \otimes E \\ \text{Id} \otimes \prec \downarrow & & \downarrow \prec \\ E \otimes E & \xrightarrow{\succ} & E \end{array}, \quad \begin{array}{ccc} E \otimes E \otimes E & \xrightarrow{\star \otimes \text{Id}} & E \otimes E \\ \text{Id} \otimes \succ \downarrow & & \downarrow \succ \\ E \otimes E & \xrightarrow{\succ} & E \end{array}.$$

Ainsi par dualité, on obtient une définition des *coalgèbre dendriforme* (introduite dans [Foi07] par Loïc Foissy) :

**Définition 11.3.3 ([Foi07, Définition 2]) :** Une *coalgèbre dendriforme*  $E$  est un  $\mathbb{k}$ -espace vectoriel  $E$  muni de deux opérations unaires :

$$\begin{aligned}\Delta_{\prec} : E &\longrightarrow E \otimes E, \\ \Delta_{\succ} : E &\longrightarrow E \otimes E,\end{aligned}$$

qui satisfont les axiomes :

$$(\Delta_{\prec} \otimes \text{Id}) \circ \Delta_{\prec}(x) = (\text{Id} \otimes \Delta_{\prec} + \text{Id} \otimes \Delta_{\succ}) \circ \Delta_{\prec}(x), \quad (11.5)$$

$$(\Delta_{\succ} \otimes \text{Id}) \circ \Delta_{\prec}(x) = (\text{Id} \otimes \Delta_{\prec}) \circ \Delta_{\succ}(x), \quad (11.6)$$

$$(\Delta_{\prec} \otimes \text{Id} + \Delta_{\succ} \otimes \text{Id}) \circ \Delta_{\succ}(x) = (\text{Id} \otimes \Delta_{\succ}) \circ \Delta_{\succ}(x), \quad (11.7)$$

pour tout  $x \in E$ .

On appelle *coproduit gauche* et *coproduit droit* respectivement  $\Delta_{\prec}$  et  $\Delta_{\succ}$ . On étend la notation de Sweedler (cf. Remarque 9.2.3) aux coalgèbres dendriformes :

$$\Delta_{\prec}(x) = x_{\prec}^L \otimes x_{\prec}^R \quad \text{et} \quad \Delta_{\succ}(x) = x_{\succ}^L \otimes x_{\succ}^R.$$

**Définition 11.3.4 ([Foi07, Définition 4]) :** Une *bialgèbre codendriforme*  $E$  est un  $\mathbb{k}$ -espace vectoriel muni de

- une structure d'algèbre (non-unitaire) avec le produit  $\mu$ ,
- une structure de coalgèbre dendriforme avec les coproduits  $\Delta_{\prec}$  et  $\Delta_{\succ}$ ,

telles que les axiomes suivants sont satisfaits :

$$\Delta_{\succ}(ab) = a^L b_{\succ}^L \otimes a^R b_{\succ}^R + a^L \otimes a^R b + ab_{\succ}^L \otimes b_{\succ}^R + b_{\succ}^L \otimes ab_{\succ}^R + a \otimes b, \quad (11.8)$$

$$\Delta_{\prec}(ab) = a^L b_{\prec}^L \otimes a^R b_{\prec}^R + a^L b \otimes a^R + ab_{\prec}^L \otimes b_{\prec}^R + b_{\prec}^L \otimes ab_{\prec}^R + b \otimes a, \quad (11.9)$$

pour tous  $a, b \in E$ .

**Définition 11.3.5 ([Foi07, Définition 9]) :** Une *bialgèbre bidendriforme*  $E$  est un  $\mathbb{k}$ -espace vectoriel muni de

- une structure d'algèbre dendriforme avec les produits gauche  $\prec$  et droit  $\succ$ ,
- une structure de coalgèbre dendriforme avec les coproduits gauche  $\Delta_{\prec}$  et droit  $\Delta_{\succ}$ ,

telles que les axiomes suivants sont satisfaits :

$$\Delta_{\succ}(a \succ b) = a^L b_{\succ}^L \otimes a^R \succ b_{\succ}^R + a^L \otimes a^R \succ b + ab_{\succ}^L \otimes b_{\succ}^R + b_{\succ}^L \otimes a \succ b_{\succ}^R + a \otimes b, \quad (11.10)$$

$$\Delta_{\succ}(a \prec b) = a^L b_{\succ}^L \otimes a^R \prec b_{\succ}^R + a^L \otimes a^R \prec b + b_{\succ}^L \otimes a \prec b_{\succ}^R, \quad (11.11)$$

$$\Delta_{\prec}(a \succ b) = a^L b_{\prec}^L \otimes a^R \succ b_{\prec}^R + ab_{\prec}^L \otimes b_{\prec}^R + b_{\prec}^L \otimes a \succ b_{\prec}^R, \quad (11.12)$$

$$\Delta_{\prec}(a \prec b) = a^L b_{\prec}^L \otimes a^R \prec b_{\prec}^R + a^L b \otimes a^R + b_{\prec}^L \otimes a \prec b_{\prec}^R + b \otimes a, \quad (11.13)$$

pour tous  $a, b \in E$ .

### 11.3.2. Coalgèbre dendriforme

Les coproduits des algèbres de Hopf définis à partir d'applications partitionnantes se définissent à partir de l'opérateur de démélange  $\delta_{\mathcal{A} \rightleftharpoons \mathcal{B}}$  (10.7).

On découpe  $\delta_{\mathcal{A} \rightleftharpoons \mathcal{B}}$  en trois opérateurs respectivement *démélange primitif*, *démélange gauche* et *démélange droit* :

$$\begin{aligned} \delta_{\mathcal{A} \rightleftharpoons \mathcal{B}}^P : w &\longmapsto \begin{cases} w \otimes \epsilon & \text{si } w \in \mathcal{A}^*, \\ \epsilon \otimes w & \text{si } w \in \mathcal{B}^*, \\ 0 & \text{sinon;} \end{cases} \\ \delta_{\mathcal{A} \rightleftharpoons \mathcal{B}}^{\prec} : w &\longmapsto \begin{cases} w|_{\mathcal{A}} \otimes w|_{\mathcal{B}} & \text{si la dernière lettre de } w \text{ est dans } \mathcal{A} \text{ et } w|_{\mathcal{B}} \neq \epsilon \\ 0 & \text{sinon;} \end{cases} \\ \delta_{\mathcal{A} \rightleftharpoons \mathcal{B}}^{\succ} : w &\longmapsto \begin{cases} w|_{\mathcal{A}} \otimes w|_{\mathcal{B}} & \text{si la dernière lettre de } w \text{ est dans } \mathcal{B} \text{ et } w|_{\mathcal{A}} \neq \epsilon \\ 0 & \text{sinon;} \end{cases} \end{aligned}$$

quelques soient les alphabets  $\mathcal{A}$  et  $\mathcal{B}$  et  $w \in (\mathcal{A} \oplus \mathcal{B})^*$ .

**Remarque 11.3.6 :**  $\delta_{\mathcal{A} \rightleftharpoons \mathcal{B}} = \delta_{\mathcal{A} \rightleftharpoons \mathcal{B}}^P + \delta_{\mathcal{A} \rightleftharpoons \mathcal{B}}^{\prec} + \delta_{\mathcal{A} \rightleftharpoons \mathcal{B}}^{\succ}$ .

**Exemple 11.3.7 :**

	$\delta_{\mathbb{N} \rightleftharpoons \mathbb{N}}^P$	$\delta_{\mathbb{N} \rightleftharpoons \mathbb{N}}^{\prec}$	$\delta_{\mathbb{N} \rightleftharpoons \mathbb{N}}^{\succ}$
121	121 $\otimes$ $\epsilon$	0	0
122189	$\epsilon \otimes$ 122189	0	0
8412221	0	8421 $\otimes$ 122	0
2189122	0	0	289 $\otimes$ 1122

**Remarque 11.3.8 :** Les cas où le mot vide apparaît dans le tenseur ( $\epsilon \otimes w$  ou  $w \otimes \epsilon$ ) sont traités à part afin de pouvoir manipuler les opérateurs de démélange gauche et droit dans l'idéal d'augmentation  $\mathbb{k}\mathcal{A}^+$ . Autrement dit, les opérateurs de démélange  $\delta_{\mathcal{A} \rightleftharpoons \mathcal{B}}^{\prec}$  et  $\delta_{\mathcal{A} \rightleftharpoons \mathcal{B}}^{\succ}$  sont bien définis sur l'algèbre (non unitaire) des mots non-vides.

Comme les sous-mots  $w|_{\mathcal{A}}$  et  $w|_{\mathcal{B}}$  sont complémentaires, il existe une unique permutation  $\sigma \in \text{Sh}_{m,n}$  (avec  $m := |w|_{\mathcal{A}}|$  et  $n := |w|_{\mathcal{B}}|$ ) telle que  $(w|_{\mathcal{A}}w|_{\mathcal{B}}) \cdot \sigma = w$ . On note  $\sigma_w$  cette unique permutation associée à  $w$ .

Par définition de  $\text{Sh}_{m,n}$  (Définition 1.1.20), ces définitions des démélanges gauche et droit sont équivalentes à :

$$\begin{aligned} \delta_{\mathcal{A} \rightleftharpoons \mathcal{B}}^{\prec}(w) &= \begin{cases} w|_{\mathcal{A}} \otimes w|_{\mathcal{B}} & \text{si } \sigma_w(n) = |w|_{\mathcal{A}}|, \\ 0 & \text{sinon;} \end{cases} \\ \delta_{\mathcal{A} \rightleftharpoons \mathcal{B}}^{\succ}(w) &= \begin{cases} w|_{\mathcal{A}} \otimes w|_{\mathcal{B}} & \text{si } \sigma_w(n) = |w|, \\ 0 & \text{sinon.} \end{cases} \end{aligned}$$

On pose alors

$$\text{Sh}_{m,n}^< := \{\sigma \in \text{Sh}_{m,n} \mid \sigma(m+n) = m+n\} \quad \text{et} \quad \text{Sh}_{m,n}^> := \{\sigma \in \text{Sh}_{m,n} \mid \sigma(m+n) = m\}.$$

Soit  $\phi$  une **Alph**-application partitionnante et soit  $\mathcal{H}_\phi(\mathcal{A})^+$  l'idéal d'augmentation de  $\mathcal{H}_\phi(\mathcal{A})$  induit par l'idéal d'augmentation  $\mathbb{k}\mathcal{A}^+$ . À partir du Corollaire 10.3.16, on déduit immédiatement :

**Proposition 11.3.9 :** *Les opérateurs de démélages gauche et droit sont biens définis de  $\mathcal{H}_\phi(\mathcal{A} \oplus \mathcal{B})^+$  dans  $\mathcal{H}_\phi(\mathcal{A})^+ \otimes \mathcal{H}_\phi(\mathcal{B})^+$  et les coproduits gauche et droit sont donnés par la formule :*

$$\delta_{\mathcal{A} \oplus \mathcal{B}}^*(m_\gamma(\mathcal{A} \oplus \mathcal{B})) = \sum_{\substack{i+j=k \\ i,j>0}} \sum_{\sigma \in \text{Sh}_{i,j}^*} \sum_{\substack{\tau \in \mathcal{C}_i, \nu \in \mathcal{C}_j \\ [\gamma:(\tau,\nu),\sigma]_{(\mathcal{A} \oplus \mathcal{B})} \neq \emptyset}} m_\tau(\mathcal{A}) \otimes m_\nu(\mathcal{B}), \quad (11.14)$$

où  $\ast = < \text{ ou } >$ , pour tout  $\gamma \in \mathcal{C}_k$  (avec  $k > 0$ ).

On vérifie facilement les axiomes suivants (analogues à (11.5), (11.6) et (11.7)) :

$$\begin{aligned} (\delta_{\mathcal{A} \oplus \mathcal{B}}^< \otimes \text{Id}) \circ \delta_{(\mathcal{A} \oplus \mathcal{B}) \Rightarrow \mathcal{C}}^<(w) &= (\text{Id} \otimes \delta_{\mathcal{B} \Rightarrow \mathcal{C}}^< + \text{Id} \otimes \delta_{\mathcal{B} \Rightarrow \mathcal{C}}^>) \circ \delta_{\mathcal{A} \Rightarrow (\mathcal{B} \oplus \mathcal{C})}^<(w), \\ (\delta_{\mathcal{A} \oplus \mathcal{B}}^> \otimes \text{Id}) \circ \delta_{(\mathcal{A} \oplus \mathcal{B}) \Rightarrow \mathcal{C}}^>(w) &= (\text{Id} \otimes \delta_{\mathcal{B} \Rightarrow \mathcal{C}}^<) \circ \delta_{\mathcal{A} \Rightarrow (\mathcal{B} \oplus \mathcal{C})}^>(w), \\ (\delta_{\mathcal{A} \oplus \mathcal{B}}^< \otimes \text{Id} + \delta_{\mathcal{A} \oplus \mathcal{B}}^> \otimes \text{Id}) \circ \delta_{(\mathcal{A} \oplus \mathcal{B}) \Rightarrow \mathcal{C}}^>(w) &= (\text{Id} \otimes \delta_{\mathcal{B} \Rightarrow \mathcal{C}}^>) \circ \delta_{\mathcal{A} \Rightarrow (\mathcal{B} \oplus \mathcal{C})}^>(w). \end{aligned}$$

Le premier axiome correspond à l'égalité entre tous les démélages possibles tels que le résultat est  $w|_{\mathcal{A}} \otimes w|_{\mathcal{B}} \otimes w|_{\mathcal{C}}$  si la dernière lettre de  $w$  appartient à  $\mathcal{A}$  et 0 sinon, le second correspond au cas où elle appartient à  $\mathcal{B}$ , et le troisième au cas où elle appartient à  $\mathcal{C}$ .

De ces axiomes, des formules (11.14) ainsi que de la Proposition 10.3.18 d'« indépendance » vis-à-vis de l'alphabet, on déduit le lemme suivant :

**Lemme 11.3.10 :** *Soit  $\mathcal{H}_\phi^+$  l'idéal d'augmentation de  $\mathcal{H}_\phi$ . Alors  $\mathcal{H}_\phi^+$  est muni d'une structure de coalgèbre dendriforme pour les coproduits gauche et droit :*

$$\begin{aligned} \Delta_{<} : \mathbb{k}\mathcal{C} &\xrightarrow{f^{\mathcal{A} \oplus \mathcal{B}}} \mathcal{H}_\phi(\mathcal{A} \oplus \mathcal{B}) \xrightarrow{\delta_{\mathcal{A} \oplus \mathcal{B}}^<} \mathcal{H}_\phi(\mathcal{A}) \otimes \mathcal{H}_\phi(\mathcal{B}) \xrightarrow{f^{\mathcal{A}-1} \otimes f^{\mathcal{B}-1}} \mathbb{k}\mathcal{C} \otimes \mathbb{k}\mathcal{C}, \\ \Delta_{>} : \mathbb{k}\mathcal{C} &\xrightarrow{f^{\mathcal{A} \oplus \mathcal{B}}} \mathcal{H}_\phi(\mathcal{A} \oplus \mathcal{B}) \xrightarrow{\delta_{\mathcal{A} \oplus \mathcal{B}}^>} \mathcal{H}_\phi(\mathcal{A}) \otimes \mathcal{H}_\phi(\mathcal{B}) \xrightarrow{f^{\mathcal{A}-1} \otimes f^{\mathcal{B}-1}} \mathbb{k}\mathcal{C} \otimes \mathbb{k}\mathcal{C}. \end{aligned}$$

**Corollaire 11.3.11 :**  $\Delta = \Delta_{<} + \Delta_{>}$ .

**Remarque 11.3.12 :** On peut montrer plus facilement ce lemme à partir de la formule du Corollaire 10.3.16. Cela étant, dans la sous-section suivante, on montre que l'idéal d'augmentation de telles algèbres de Hopf est automatiquement muni d'une structure de bialgèbre codendriforme. Les axiomes sont difficiles à montrer à partir de la structure  $\mathcal{H}_\phi^+$  mais la démonstration devient très simple à partir des réalisations.

**Remarque 11.3.13 :** On peut découper de manière analogue l'ensemble des permutations  $\text{Sh}_{m,n}$  avec les conditions  $\sigma(1) = 1$  ou  $\sigma(1) = n+1$ . En combinant les deux découpages, on obtient une partition de  $\text{Sh}_{m,n}$  en quatre parts. En utilisant ces quatre parts pour définir quatre coproduits (comme dans la formule (11.14)), on munit  $\mathcal{H}_\phi^+$  d'une structure quadri-coalgèbre (structure duale d'une quadri-algèbre [AL04]).

### 11.3.3. Bialgèbre codendriforme

On montre que  $\mathcal{H}_\phi^+$  est munie d'une structure de bialgèbre codendriforme très simplement à partir des réalisations polynomiales. En effet sur les mots (non-vides), on a :

**Lemme 11.3.14 :** Les opérateurs de démélange gauche  $\delta_{\mathcal{A} \oplus \mathcal{B}}^{\prec}$  et droit  $\delta_{\mathcal{A} \oplus \mathcal{B}}^{\succ}$  satisfont les axiomes (11.8) et (11.9).

PREUVE : Soient  $u$  et  $v$  deux mots de longueur non-nulle sur  $\mathcal{A} \oplus \mathcal{B}$ . Pour chaque combinaison de contraintes, on détaille dans un tableau tous les résultats possibles des coproduits gauche et droit en fonction des lettres des mots  $u$  et  $v$ , s'ils contiennent ou non des lettres de uniquement  $\mathcal{A}$ , uniquement  $\mathcal{B}$  ou les deux  $\mathcal{A} \oplus \mathcal{B}$ .

(Éq. 11.8) Pour  $\delta_{\mathcal{A} \oplus \mathcal{B}}^{\succ}(uv)$ , les différents résultats sont

$u \setminus v$	$\mathcal{A}$	$\mathcal{B}$	$\mathcal{A} \oplus \mathcal{B}$
$\mathcal{A}$	0	$u \otimes v$	$uv _{\mathcal{A}} \otimes v _{\mathcal{B}}$
$\mathcal{B}$	0	0	$v _{\mathcal{A}} \otimes uv _{\mathcal{B}}$
$\mathcal{A} \oplus \mathcal{B}$	0	$u _{\mathcal{A}} \otimes u _{\mathcal{B}}v$	$u _{\mathcal{A}}v _{\mathcal{A}} \otimes u _{\mathcal{B}}v _{\mathcal{B}}$

Dans chacun des cas, le démélange droit dépend uniquement de la dernière lettre de  $v$ , l'opérateur agit comme le démélange usuel sur  $u$ . L'axiome

$$\delta_{\mathcal{A} \oplus \mathcal{B}}^{\succ}(uv) = u^L v_{\prec}^L \otimes u^R v_{\prec}^R + u^L \otimes u^R v + uv_{\prec}^L \otimes v_{\prec}^R + v_{\prec}^L \otimes uv_{\prec}^R + u \otimes v, \quad (11.8)$$

est donc satisfait.

(Éq. 11.9) Pour  $\delta_{\mathcal{A} \oplus \mathcal{B}}^{\prec}(uv)$ , les différents résultats sont

$u \setminus v$	$\mathcal{A}$	$\mathcal{B}$	$\mathcal{A} \oplus \mathcal{B}$
$\mathcal{A}$	0	0	$uv _{\mathcal{A}} \otimes v _{\mathcal{B}}$
$\mathcal{B}$	$v \otimes u$	0	$v _{\mathcal{A}} \otimes uv _{\mathcal{B}}$
$\mathcal{A} \oplus \mathcal{B}$	$u _{\mathcal{A}}v \otimes u_{\mathcal{B}}$	0	$u _{\mathcal{A}}v _{\mathcal{A}} \otimes u _{\mathcal{B}}v _{\mathcal{B}}$

L'axiome

$$\delta_{\mathcal{A} \oplus \mathcal{B}}^{\prec}(uv) = u^L v_{\prec}^L \otimes u^R v_{\prec}^R + u^L v \otimes u^R + uv_{\prec}^L \otimes v_{\prec}^R + v_{\prec}^L \otimes uv_{\prec}^R + v \otimes u, \quad (11.9)$$

est donc satisfait. ■

Comme précédemment, comme les axiomes sont satisfaits sur les mots, ils sont automatiquement satisfaits dans la sous-algèbre  $\mathcal{H}_\phi(\mathcal{A} \oplus \mathcal{B})$ , et on en déduit qu'ils sont vérifiés aussi sur  $\mathcal{H}_\phi$ .

**Théorème 11.3.15 :** L'idéal d'augmentation  $\mathcal{H}_\phi^+$  est une bialgèbre codendriforme.

PREUVE : D'après le Lemme 11.3.14, les opérateurs de démélange satisfont les axiomes (11.8) et (11.9), on déduit automatiquement du Lemme 11.3.10 que la coalgèbre dendriforme  $\mathcal{H}_\phi^+$  (isomorphe à  $\mathcal{H}_\phi(\mathcal{A})^+$ ) est satisfait aussi les axiomes. ■



### 11.3.4. Bialgèbre bidendriforme

De manière générale, il n'existe pas de structure dendriforme sur l'algèbre libre d'un alphabet quelconque  $\mathcal{A}$ . Il n'est donc pas possible de déduire des propriétés de bialgèbre (co/bi-)dendriforme en toute généralité.

Dans [NT07], Jean-Christophe NOVELLI et Jean-Yves THIBON se sont intéressés au cas des alphabets totalement ordonnés. Ils associent une structure (tri-)dendriforme à l'idéal d'augmentation  $\mathbb{k}\langle\mathcal{A}\rangle^+$  de l'algèbre libre sur un tel alphabet  $\mathcal{A}$  :

**Définition 11.3.16** ([NT07, Définition 2.1]) : Soient  $u$  et  $v$  deux mots non-vides sur l'alphabet  $\mathcal{A}$ .

$$\begin{aligned} u \ll v &:= \begin{cases} uv & \text{si } \max(u) > \max(v), \\ 0 & \text{sinon;} \end{cases} \\ u \circ v &:= \begin{cases} uv & \text{si } \max(u) = \max(v), \\ 0 & \text{sinon;} \end{cases} \\ u \gg v &:= \begin{cases} uv & \text{si } \max(u) < \max(v), \\ 0 & \text{sinon;} \end{cases} \end{aligned}$$

On définit le produit gauche et le produit droit respectivement par :

$$\prec := \ll \quad \text{et} \quad \succ := \circ + \gg .$$

**Proposition 11.3.17** :  $(\mathbb{k}\mathcal{A}^+, \prec, \succ)$  est une algèbre dendriforme.

PREUVE : Par définition,  $(u \prec v) \prec w = uvw$  si et seulement si  $\max(u) > \max(v)$  et  $\max(uv) > \max(w)$ . Ceci est équivalent à  $\max(u) > \max(vw)$ . Par définition  $v \prec w + v \succ w = vw$ , et on a  $u \prec vw = uvw$  si et seulement si  $\max(u) > \max(vw)$ . L'axiome (11.2) est donc satisfait.

En appliquant le même raisonnement pour les axiomes (11.3) et (11.4), on montre que  $(\mathbb{k}\mathcal{A}^+, \prec, \succ)$  est une algèbre dendriforme. ■

**Remarque 11.3.18** : En introduisant la notion d'algèbre tridendriforme [LR04, § 2.1] sous-jacente à la Définition 11.3.16, il suffit de combiner certains axiomes tridendriformes pour retrouver les axiomes dendriformes.

À partir de cette définition d'algèbre dendriforme sur un alphabet totalement ordonné  $\mathcal{A}$ , on s'intéresse à savoir sous quelles conditions les produits gauche et droit stabilisent l'espace  $\mathcal{H}_\phi(\mathcal{A})$ .

Soit  $\gamma$  une structure de degré  $n \geq 1$ . Pour tout  $1 \leq i < n$ , on pose :

$$\begin{aligned} \mathbb{1}_{\prec}^{\gamma}(i) &:= \{w \in \phi_{\mathcal{A}}^{-1}(\gamma) \mid w = w[1, i] \prec w[i+1, n]\}, \\ \mathbb{1}_{\succ}^{\gamma}(i) &:= \{w \in \phi_{\mathcal{A}}^{-1}(\gamma) \mid w = w[1, i] \succ w[i+1, n]\}. \end{aligned}$$

**Lemme 11.3.19 :** Les produits gauche et droit stabilisent  $\mathcal{H}_\phi(\mathcal{A})^+$  si et seulement si pour tout  $n \geq 1$ , toute structure  $\gamma \in \mathcal{C}_n$  et tout  $1 \leq i < n$ ,

$$\mathbb{1}_\gamma(i) = \phi_{\mathcal{A}}^{-1}(\gamma) \quad \text{ou} \quad \mathbb{1}_\gamma(i) = \phi_{\mathcal{A}}^{-1}(\gamma). \quad (11.15)$$

PREUVE : Le raisonnement est identique à celui du Lemme 10.3.4. ■

Si la contrainte (11.15) est satisfaite, on dit que  $\phi$  est une *application partitionnante dendriforme*.

Comme précédemment avec les autres opérateurs, on enrichit  $\mathcal{H}_\phi^+$  d'une structure dendriforme :

$$\begin{aligned} \mu_{\prec} : \mathbb{k}\mathcal{C} \otimes \mathbb{k}\mathcal{C} &\xrightarrow{f^{\mathcal{A}} \otimes f^{\mathcal{A}}} \mathcal{H}_\phi(\mathcal{A}) \otimes \mathcal{H}_\phi(\mathcal{A}) \xrightarrow{\prec} \mathcal{H}_\phi(\mathcal{A}) \xrightarrow{f^{\mathcal{A}-1}} \mathbb{k}\mathcal{C}, \\ \mu_{\succ} : \mathbb{k}\mathcal{C} \otimes \mathbb{k}\mathcal{C} &\xrightarrow{f^{\mathcal{A}} \otimes f^{\mathcal{A}}} \mathcal{H}_\phi(\mathcal{A}) \otimes \mathcal{H}_\phi(\mathcal{A}) \xrightarrow{\succ} \mathcal{H}_\phi(\mathcal{A}) \xrightarrow{f^{\mathcal{A}-1}} \mathbb{k}\mathcal{C}. \end{aligned}$$

Dans la précédente sous-section, on a montré que  $\mathcal{H}_\phi^+$  est une bialgèbre codendriforme (Théorème 11.3.15). On suppose maintenant que  $\mathcal{H}_\phi^+$  est munie d'une telle structure d'algèbre dendriforme. On a alors que :

**Théorème 11.3.20 :**  $\mathcal{H}_\phi^+$  est une *bialgèbre bidendriforme*.

PREUVE : Par hypothèse  $\mathcal{H}_\phi^+$  est une algèbre dendriforme, et d'après le Théorème 11.3.15, c'est aussi une bialgèbre codendriforme (donc en particulier une coalgèbre dendriforme). On montre que les axiomes (11.10) à (11.13) des bialgèbres bidendriformes sont vérifiés au niveau des opérateurs de démixage gauche et droit sur les mots et on déduit comme précédemment que  $\mathcal{H}_\phi^+$  est ainsi munie d'une structure de bialgèbre bidendriforme.

Soient  $u$  et  $v$  deux mots de longueur non-nulle sur  $\mathcal{A} \oplus \mathcal{B}$ . Pour chaque combinaison de contraintes, on détaille dans un tableau tous les résultats possibles des coproduits gauche et droit en fonction des lettres des mots  $u$  et  $v$ , s'ils contiennent ou non des lettres de uniquement  $\mathcal{A}$ , uniquement  $\mathcal{B}$  ou les deux  $\mathcal{A} \oplus \mathcal{B}$ . On note  $-$  lorsque les contraintes sont incohérentes.

[Éq. 11.10] Sous l'hypothèse  $\max(u) \leq \max(v)$  (c'est-à-dire  $u \succ v = uv$ ), les différents résultats possibles de  $\delta_{\mathcal{A} \oplus \mathcal{B}}^\succ(u \succ v)$  sont

$u \backslash v$	$\mathcal{A}$	$\mathcal{B}$	$\mathcal{A} \oplus \mathcal{B}$
$\mathcal{A}$	0	$u \otimes v$	$uv _{\mathcal{A}} \otimes v _{\mathcal{B}}$
$\mathcal{B}$	$-$	0	$v _{\mathcal{A}} \otimes uv _{\mathcal{B}}$
$\mathcal{A} \oplus \mathcal{B}$	$-$	$u _{\mathcal{A}} \otimes u _{\mathcal{B}}v$	$u _{\mathcal{A}}v _{\mathcal{A}} \otimes u _{\mathcal{B}}v _{\mathcal{B}}$

Par hypothèse,

$$u|_{\mathcal{B}}v = u|_{\mathcal{B}} \succ v, \quad uv|_{\mathcal{B}} = u \succ v|_{\mathcal{B}} \quad \text{et} \quad u|_{\mathcal{B}}v|_{\mathcal{B}} = u|_{\mathcal{B}} \succ v|_{\mathcal{B}}.$$

Pour les mêmes raisons que dans la preuve du Lemme 11.3.14, on déduit que l'axiome :

$$\delta_{\mathcal{A} \oplus \mathcal{B}}^{\succ}(u \succ v) = u^L v_{\prec}^L \otimes u^R v_{\prec}^R + u^L \otimes u^R v + uv_{\prec}^L \otimes v_{\prec}^R + v_{\prec}^L \otimes uv_{\prec}^R + u \otimes v, \quad (11.10)$$

est satisfait. Si  $\max(u) > \max(v)$  alors  $\delta_{\mathcal{A} \oplus \mathcal{B}}^{\succ}(u \succ v) = \delta_{\mathcal{A} \oplus \mathcal{B}}^{\succ}(0) = 0$  et l'axiome est donc satisfait aussi. (Il en va de même pour les axiomes suivants, si l'hypothèse est contredite l'opération vaut 0 est donc l'axiome est satisfait.)

[Éq. 11.11] Sous l'hypothèse  $\max(u) > \max(v)$  (c'est-à-dire  $u \succ v = uv$ ), les différents résultats possibles de  $\delta_{\mathcal{A} \oplus \mathcal{B}}^{\succ}(u \prec v)$  sont

$u \setminus v$	$\mathcal{A}$	$\mathcal{B}$	$\mathcal{A} \oplus \mathcal{B}$
$\mathcal{A}$	0	—	—
$\mathcal{B}$	0	0	$v_{ \mathcal{A}} \otimes uv_{ \mathcal{B}}$
$\mathcal{A} \oplus \mathcal{B}$	0	$u_{ \mathcal{A}} \otimes u_{ \mathcal{B}} v$	$u_{ \mathcal{A}} v_{ \mathcal{A}} \otimes u_{ \mathcal{B}} v_{ \mathcal{B}}$

Par hypothèse,

$$u_{|\mathcal{B}} v = u_{|\mathcal{B}} \prec v, \quad uv_{|\mathcal{B}} = u \prec v_{|\mathcal{B}} \quad \text{et} \quad u_{|\mathcal{B}} v_{|\mathcal{B}} = u_{|\mathcal{B}} \prec v_{|\mathcal{B}}.$$

On obtient donc que l'axiome :

$$\delta_{\mathcal{A} \oplus \mathcal{B}}^{\prec}(u \prec v) = u^L v_{\prec}^L \otimes u^R \prec v_{\prec}^R + u^L \otimes u^R \prec v + v_{\prec}^L \otimes u \prec v_{\prec}^R, \quad (11.11)$$

est satisfait.

[Éq. 11.12] Sous l'hypothèse  $\max(u) \leq \max(v)$ , les différents résultats possibles de  $\delta_{\mathcal{A} \oplus \mathcal{B}}^{\prec}(u \succ v)$  sont

$u \setminus v$	$\mathcal{A}$	$\mathcal{B}$	$\mathcal{A} \oplus \mathcal{B}$
$\mathcal{A}$	0	0	$uv_{ \mathcal{A}} \otimes v_{ \mathcal{B}}$
$\mathcal{B}$	—	0	$v_{ \mathcal{A}} \otimes uv_{ \mathcal{B}}$
$\mathcal{A} \oplus \mathcal{B}$	—	0	$u_{ \mathcal{A}} v_{ \mathcal{A}} \otimes u_{ \mathcal{B}} v_{ \mathcal{B}}$

Par hypothèse,

$$uv_{|\mathcal{B}} = u \succ v_{|\mathcal{B}} \quad \text{et} \quad u_{|\mathcal{B}} v_{|\mathcal{B}} = u_{|\mathcal{B}} \succ v_{|\mathcal{B}}.$$

L'axiome :

$$\delta_{\mathcal{A} \oplus \mathcal{B}}^{\prec}(u \succ v) = u^L v_{\prec}^L \otimes u^R \succ v_{\prec}^R + uv_{\prec}^L \otimes v_{\prec}^R + v_{\prec}^L \otimes u \succ v_{\prec}^R, \quad (11.12)$$

est satisfait.

[Éq. 11.13] Sous l'hypothèse  $\max(u) > \max(v)$ , les différents résultats possibles de  $\delta_{\mathcal{A} \oplus \mathcal{B}}^{\prec}(u \prec v)$  sont

$u \setminus v$	$\mathcal{A}$	$\mathcal{B}$	$\mathcal{A} \oplus \mathcal{B}$
$\mathcal{A}$	0	—	—
$\mathcal{B}$	$v \otimes u$	0	$v_{ \mathcal{A}} \otimes uv_{ \mathcal{B}}$
$\mathcal{A} \oplus \mathcal{B}$	$u_{ \mathcal{A}} v \otimes u_{ \mathcal{B}}$	0	$u_{ \mathcal{A}} v_{ \mathcal{A}} \otimes u_{ \mathcal{B}} v_{ \mathcal{B}}$

Par hypothèse,

$$uv|_{\mathcal{B}} = u \prec v|_{\mathcal{B}} \quad \text{et} \quad u|_{\mathcal{B}}v|_{\mathcal{B}} = u|_{\mathcal{B}} \prec v|_{\mathcal{B}}.$$

L'axiome

$$\delta_{\mathcal{A} \rightleftharpoons \mathcal{B}}^{\prec}(u \prec v) = u^L v_{\prec}^L \otimes u^R \prec v_{\prec}^R + u^L v \otimes u^R + v_{\prec}^L \otimes u \prec v_{\prec}^R + v \otimes u, \quad (11.13)$$

est satisfait.

Les axiomes de bialgèbre bidendriforme sont satisfaits sur le  $\mathbb{k}$ -espace vectoriel  $\mathbb{k}\mathcal{A}^+$  quelque soit l'alphabet  $\mathcal{A}$ . Comme les opérateurs sont bien définis sur le sous-espace  $\mathcal{H}_{\phi}(\mathcal{A})^+$ , on déduit que ces axiomes sont vérifiés aussi sur cet espace. Enfin à partir des propriétés d'indépendance vis-à-vis de l'alphabet, on déduit alors que  $\mathcal{H}_{\phi}^+$  est une bialgèbre bidendriforme. ■

Ce dernier théorème donne un cadre très large. Il permet de déduire la structure de bialgèbre bidendriforme des idéaux d'augmentations des algèbres de Hopf immédiatement à partir des propriétés des applications partitionnantes.

## 12. Exemples de réalisations

Dans ce chapitre, pour conclure la partie sur les applications partitionnantes et les algèbres de Hopf, on explicite la définition de l'algèbre de Hopf **FQSym** [MR95 ; DHT02] à partir de l'application partitionnante *standardisation*. On introduit aussi une nouvelle application partitionnante qui généralise la *parkisation* introduite par Jean-Christophe NOVELLI et Jean-Yves THIBON comme l'application partitionnante qui définit l'algèbre de Hopf des fonctions de parking. Cette généralisation définit une application qui associe, à tout mot sur un alphabet avec différence, une  $\chi$ -fonction de parking. On étudie et montre que cette application est partitionnante si et seulement si la fonction discrète  $\chi$  est de la forme  $\chi : n \mapsto 1 + m(n - 1)$  avec  $m \geq 0$  entier.

### 12.1. Fonctions quasi-symétriques libres, FQSym

L'algèbre de Hopf des *fonctions quasi-symétriques libres*, **FQSym**, a été introduite par Claudia MALVENUTO et Christophe REUTENAUER dans [MR95]. Dans [DHT02], Gérard H. E. DUCHAMP, Florent HIVERT et Jean-Yves THIBON réintroduisent cette algèbre de Hopf en utilisant une application partitionnante : la *standardisation*.

Cette algèbre de Hopf combinatoire est indexée par les permutations et est l'une des seules algèbres de Hopf autoduales dont on connaît un morphisme explicite avec sa duale.

#### 12.1.1. Alphabets totalement ordonnés et standardisation

Avant d'introduire la standardisation, on introduit les *ensembles d'inversions* des séquences ordonnées :

**Définition 12.1.1 :** Soit  $A := (a_1, a_2, \dots, a_m)$  une séquence (mot, ordre linéaire, permutation, etc.) d'éléments  $a_i$  sur un alphabet muni d'un ordre total  $<_a$ . L'*ensemble d'inversion* de  $A$  est l'ensemble :

$$\text{Inv}(A) := \{(i, j) \mid i < j \text{ et } a_j <_a a_i \text{ et } a_j \neq a_i\}.$$

**Exemple 12.1.2 :**

$$\text{Inv}(14232155) = \{(2, 3), (2, 4), (2, 5), (2, 6), (4, 5), (4, 6)\}$$

Soit  $\text{Ens}^\infty$  le type des alphabets totalement ordonnés et soit  $(\mathcal{A}, <_{\mathcal{A}})$  un alphabet de ce type. Soit  $\mathcal{S}$  la classe combinatoire des permutations (§ 1.1.5).

**Définition 12.1.3 :** La **standardisation**, notée **std**, est la donnée qui associe, à tout alphabet totalement ordonné  $\mathcal{A}$ , l'application graduée :

$$\text{std}_{\mathcal{A}} : \begin{cases} \mathcal{A}^* & \longrightarrow \mathcal{S} \\ w := a_1 a_2 \cdots a_n & \longmapsto \sigma := \sigma_1 \sigma_2 \cdots \sigma_n \end{cases}$$

où  $\sigma$  est l'unique permutation ayant même ensemble d'inversions, c'est-à-dire l'unique permutation de  $\mathcal{S}_n$  telle que  $\text{Inv}(w) = \text{Inv}(\sigma)$ .

**Exemple 12.1.4 :** Soit 142322123 un mot d'entiers (strictement inférieurs à 10).

$$\text{std}_{\mathbb{N}}(142322123) = 193745268.$$

**Proposition 12.1.5 :** La donnée **std** est une  $\text{Ens}_{\prec}^{\infty}$ -application partitionnante.

PREUVE :

- Par définition, la standardisation est graduée (Condition 2).
- Par hypothèse, on considère des alphabets  $\mathcal{A}$  infinis et totalement ordonnés. L'ensemble des entiers est infini et totalement ordonné. Toute permutation  $\sigma \in \mathcal{S}_n$  est « identifiable » à un mot sur l'ensemble des entiers. Par définition de la standardisation, on a  $\text{std}_{\mathbb{N}}(\sigma) = \sigma$ . L'application  $\text{std}_{\mathbb{N}}$  est donc surjective. Plus généralement, à partir de n'importe quelle injection croissante de  $[n]$  dans  $\mathcal{A}$ , par le même raisonnement on déduit que  $\text{std}_{\mathcal{A}}$  est surjective (Condition 1).
- Par définition d'un type d'alphabet, les injections gauche et droite sont des morphismes pour l'ordre. L'image d'une injection croissante préserve l'ensemble d'inversions. Donc la standardisation commute avec gauche et droit (Condition 3).
- L'ensemble d'inversions d'un mot est un ensemble de couples de positions de symboles du mot satisfaisants une condition qui ne dépend que de ces deux symboles et de ces deux positions. D'où, si deux mots  $w$  et  $w'$  (de longueur  $n$ ) ont le même ensemble d'inversions alors, pour tout  $1 \leq i < j \leq n$ , les facteurs  $w[i, j]$  et  $w'[i, j]$  ont aussi le même ensemble d'inversions, et par conséquent le même standardisé (Condition 4).
- Soient  $u \in \mathcal{A}^m$  et  $v \in \mathcal{B}^n$  et soient  $1 \leq i < j \leq m + n$ . Par ailleurs, par définition de la somme des alphabets ordonnés, dans  $\mathcal{A} \oplus \mathcal{B}$  les symboles issus de  $\mathcal{A}$  sont strictement plus petits que ceux issus de  $\mathcal{B}$ . On en déduit que : pour toute permutation  $\sigma \in \text{Sh}_{m,n}$ ,

$$\begin{aligned} \text{Inv}((uv) \cdot \sigma) &= \text{Inv}(\sigma) \cup \\ &\quad \{(\sigma(i), \sigma(j)) \mid (i, j) \in \text{Inv}(u)\} \cup \\ &\quad \{(\sigma(i - m), \sigma(j - m)) \mid (i, j) \in \text{Inv}(v)\}. \end{aligned} \tag{12.1}$$

On en déduit que la compatibilité au mélange est satisfaite (Condition 5). La standardisation est donc une application partitionnante. ■

### 12.1.2. FQSym duale, alias $\mathcal{H}_{\text{std}}$

La standardisation étant une  $\text{Ens}_{\leq}^{\infty}$ -application partitionnante, on obtient, à partir des constructions des précédentes sections (§ 10.3 et § 10.4), deux algèbres de Hopf  $\mathcal{H}_{\text{std}}$  et  $\mathcal{H}_{\text{std}}^{\vee}$ , indexées par la classe combinatoire des permutations. La première construction,  $\mathcal{H}_{\text{std}}$ , définit l'algèbre de Hopf duale de **FQSym** dans la base fondamentale duale  $(\mathbf{G}_{\sigma})$ .

Chaque  $\mathbf{G}_{\sigma}$  est défini à partir de la fibre en  $\sigma$  de  $\text{std}$  (10.2).

**Exemple 12.1.6 :** Soit  $\sigma := 142536$  :

$$\mathbf{G}_{\sigma}(\mathbb{N}) = \sum_{\substack{w \in \mathbb{N}^* \\ \phi_{\mathbb{N}}(w) = \sigma}} w = 121212 + 121213 + 121214 + \cdots + 121313 + \cdots + 142535 + \cdots$$

Nous avons montré (§ 10.3) que la structure d'algèbre de Hopf est une conséquence des bonnes propriétés de la standardisation. En mettant de côté l'aspect algébrique (associativité, co-associativité, etc.), le produit et le coproduit définissent des règles (purements combinatoire) d'assemblage et de désassemblage des permutations.

**Définition 12.1.7 :** La **concaténation décalée**  $\tau \gamma[m]$  est la concaténation des permutations de  $\tau$  avec  $\gamma$  décalée de  $m$  :

$$\tau \gamma[m] : i \longmapsto \begin{cases} \tau(i) & \text{si } i \leq m, \\ \gamma(i - m) + m & \text{si } i > m. \end{cases}$$

Le produit dans la base  $(\mathbf{G}_{\sigma})$  se définit par la règle suivante : soient  $\tau \in \mathcal{S}_m$  et  $\nu \in \mathcal{S}_n$  deux permutations,

$$\mathbf{G}_{\tau} \mathbf{G}_{\nu} = \sum_{\sigma \in \text{Sh}_{m,n}} \mathbf{G}_{((\tau^{-1} \nu^{-1}[m]) \cdot \sigma)^{-1}}.$$

On déduit cette expression combinatoire du produit à partir des réalisations polynômes. (La suite est une brève description, sans preuve, de comment déduire cette formule combinatoire à partir des mots.)

Partant du constat que la permutation inverse de la standardisation d'un mot trie ce mot, c'est-à-dire si  $\text{std}_{\mathcal{A}}(u) = \tau$  alors  $u^{\uparrow} = u \cdot \tau^{-1}$ . À la manière (de la dernière étape) du tri fusion, pour tous mots  $u$  et  $v$  (respectivement de longueur  $m$  et  $n$ ), il existe une unique permutation  $\sigma$  de  $\text{Sh}_{m,n}$  qui effectue un tri stable des triés de  $u$  et  $v$ , c'est-à-dire telle que

$$(u^{\uparrow} v^{\uparrow}) \cdot \sigma = (uv)^{\uparrow}$$

est un tri stable de  $u^{\uparrow}$  et  $v^{\uparrow}$ .

Pour  $\tau := \text{std}_{\mathcal{A}}(u)$  et  $\nu := \text{std}_{\mathcal{A}}(v)$ , à partir de la définition de la concaténation décalée, on montre alors que  $(\tau^{-1} \cdot \nu^{-1}[m]) \cdot \sigma = \text{std}_{\mathcal{A}}(uv)^{-1}$ . À chaque permutation

$\sigma$  de  $\text{Sh}_{m,n}$ , on peut toujours associer un couple de mots  $(u, v)$  tel que  $(u^\uparrow v^\uparrow) \cdot \sigma = (uv)^\uparrow$ . Par ailleurs, le Corollaire 10.3.6 nous assure que le produit est sans multiplicité, on déduit alors que :

$$\begin{aligned} \mathbf{G}_\tau(\mathcal{A})\mathbf{G}_\nu(\mathcal{A}) &= \sum_{[\tau, \nu: \gamma]_{\mathcal{A}} \neq \emptyset} \mathbf{G}_\gamma(\mathcal{A}), \\ &= \sum_{\sigma \in \text{Sh}_{m,n}} \mathbf{G}_{((\tau^{-1} \nu^{-1}[m]) \cdot \sigma)^{-1}(\mathcal{A})}. \end{aligned}$$

**Exemple 12.1.8 :**

$$\begin{aligned} \mathbf{G}_{132}\mathbf{G}_{12} &= \mathbf{G}_{13245} + \mathbf{G}_{14235} + \mathbf{G}_{14325} + \mathbf{G}_{15234} + \mathbf{G}_{15324} + \mathbf{G}_{15423} + \\ &\quad \mathbf{G}_{24315} + \mathbf{G}_{25314} + \mathbf{G}_{25413} + \mathbf{G}_{35412} \end{aligned}$$

De même pour le coproduit, on a la formule combinatoire :

$$\Delta(\mathbf{G}_\gamma) = \sum_{i=0}^n \mathbf{G}_{\gamma|_{[i]}} \otimes \mathbf{G}_{\text{std}(\gamma|_{\{i+1, \dots, n\}})}$$

avec  $\gamma|_F$  la restriction de  $\gamma$  à l'ensemble  $F$ .

Cette formule se déduit du dédoublement d'alphabet : soit  $\gamma \in \mathcal{S}_n$ ,

$$\delta_{\mathcal{A} \oplus \mathcal{B}}(\mathbf{G}_\gamma(\mathcal{A} \oplus \mathcal{B})) = \sum_{i+j=n} \sum_{\substack{\tau \in \mathcal{S}_i \\ \nu \in \mathcal{S}_j}} C_{\tau, \nu}^\gamma \mathbf{G}_\tau(\mathcal{A}) \otimes \mathbf{G}_\nu(\mathcal{B}).$$

On déduit de l'Équation 12.1 que, pour tous  $w$  et  $w'$  (tels que  $\text{std}_{\mathcal{A} \oplus \mathcal{B}}(w) = \text{std}_{\mathcal{A} \oplus \mathcal{B}}(w')$ ), si  $i := \ell(w|_{\mathcal{A}}) = \ell(w'|_{\mathcal{A}})$  alors les ensembles d'inversions de  $w|_{\mathcal{A}}$  et  $w|_{\mathcal{B}}$  sont respectivement les mêmes que ceux de  $w'|_{\mathcal{A}}$  et  $w'|_{\mathcal{B}}$ . Autrement dit, pour  $i$  fixé, il existe un unique couple de permutations  $(\tau, \nu)$  définit comme le couple des restrictions de  $\gamma$  aux ensembles  $[i]$  et  $\{i+1, \dots, n\}$ , c'est-à-dire  $\tau := \gamma|_{[i]}$  et  $\nu := \text{std}(\gamma|_{\{i+1, \dots, n\}})$ . Par ailleurs, pour tout  $i$ , on peut toujours construire un mot  $w$  tel que  $\text{std}_{\mathcal{A} \oplus \mathcal{B}}(w) = \gamma$  et  $w$  contient exactement  $i$  symboles de  $\mathcal{A}$ . D'où la formule combinatoire.

**Exemple 12.1.9 :**

$$\Delta(\mathbf{G}_{3142}) = 1 \otimes \mathbf{G}_{3142} + \mathbf{G}_1 \otimes \mathbf{G}_{231} + \mathbf{G}_{12} \otimes \mathbf{G}_{12} + \mathbf{G}_{312} \otimes \mathbf{G}_1 + \mathbf{G}_{3142} \otimes 1$$

### 12.1.3. FQSym, alias $\mathcal{H}_{\text{std}}^\vee$

À partir de la seconde construction, l'algèbre de Hopf  $\mathcal{H}_{\text{std}}^\vee$  définit **FQSym** dans la base fondamentale  $(\mathbf{F}_\sigma)$ . Le produit est simplement le produit de mélange (décalé) des permutations : soient  $\tau \in \mathcal{S}_m$  et  $\nu \in \mathcal{S}_n$ ,

$$\mathbf{F}_\tau \mathbf{F}_\nu = \sum_{\sigma \in \text{Sh}_{m,n}} \mathbf{F}_{(\tau \nu[m]) \cdot \sigma}.$$

Cette formule est immédiate à partir de la définition de la somme des alphabets ordonnés  $\mathcal{A} \oplus \mathcal{B}$  et du produit de mélange  $\sqcup_{\mathcal{A}, \mathcal{B}}$ .



**Exemple 12.1.10 :**

$$\begin{aligned} \mathbf{F}_{132}\mathbf{F}_{12} = & \mathbf{F}_{13245} + \mathbf{F}_{13425} + \mathbf{F}_{13452} + \mathbf{F}_{14325} + \mathbf{F}_{14352} + \mathbf{F}_{14532} + \\ & \mathbf{F}_{41325} + \mathbf{F}_{41352} + \mathbf{F}_{41532} + \mathbf{F}_{45132} \end{aligned}$$

De même, le coproduit de déconcaténation se traduit en la déconcaténation standardisée des permutations : soit  $\sigma \in \mathcal{S}_n$ ,

$$\Delta(\mathbf{F}_\sigma) = \sum_{i=0}^n \mathbf{F}_{\text{std}(\sigma[1,i])} \otimes \mathbf{F}_{\text{std}(\sigma[i+1,i])}.$$

**Exemple 12.1.11 :**

$$\Delta(\mathbf{F}_{3241}) = 1 \otimes \mathbf{F}_{3241} + \mathbf{F}_1 \otimes \mathbf{F}_{231} + \mathbf{F}_{21} \otimes \mathbf{F}_{21} + \mathbf{F}_{213} \otimes \mathbf{F}_1 + \mathbf{F}_{3241} \otimes 1$$

### 12.1.4. Propriété dendriforme

À partir de l'ensemble d'inversion d'un mot  $uv$  sur un alphabet totalement ordonné, on peut facilement déduire si :

$$\max(u) > \max(v) \quad \text{ou} \quad \max(u) \leq \max(v).$$

En effet, si  $\max(u) > \max(v)$ , les positions  $i$  de maximums de  $u$  définissent des inversions avec les positions  $j$  de chaque symbole de  $v$  (dans  $uv$ ),  $(i, j) \in \text{Inv}(uv)$ . Réciproquement si  $\max(u) \leq \max(v)$ , il existe au moins une position  $j$  de  $v$  (dans  $uv$ ) pour laquelle le couple  $(i, j)$  n'est pas une inversion de  $uv$ , quelque soit la position  $i$  dans  $u$ .

On déduit que la standardisation satisfait l'Équation 11.15, et rentre ainsi dans le cadre des applications partitionnantes dendriformes (§ 11.3.4). À partir du Théorème 11.3.20, on retrouve le résultat de Loïc Foissy [Foi07, Théorème 38] : l'idéal d'augmentation de  $\mathbf{FQSym}$ , alias  $\mathcal{H}_{\text{std}}^+$  est une bialgèbre bidendriforme.

## 12.2. Fonction de parking généralisée, parkisation et algèbre de Hopf

L'algèbre de Hopf des fonctions de parking,  $\mathbf{PQSym}$ , a été introduite, par Jean-Christophe NOVELLI et Jean-Yves THIBON dans [NT04a], pour étudier le lien entre certaines fonctions de Schur associées à une mesure de probabilité et le caractère de Frobenius des fonctions de parking primitives sous l'action du groupe symétrique. Tout comme  $\mathbf{FQSym}$  ou  $\mathbf{WQSym}$ , elle se définit très bien en terme de réalisations polynomiales par l'intermédiaire de la *parkisation* et son idéal d'augmentation est muni d'une structure de bialgèbre bidendriforme [NT07].

Dans [NT14], les mêmes auteurs mettent en avant toute une famille d'algèbres de Hopf sur les  $\chi$ -fonctions de parking (Chapitre 4) dans le cas où  $\chi(n) := 1 + m(n - 1)$  avec  $m \geq 1$ . Dans cette section, on explicite une généralisation de la parkisation, en fonction de  $\chi$ , et on montre que cette donnée n'est une application partitionnante que lorsque  $\chi(n) := 1 + m(n - 1)$ .

### 12.2.1. Parkisation

Avant d'introduire la parkisation, on définit les notions d'évaluation et de points de rupture d'un mot.

**Définition 12.2.1 :** Soit  $w \in \mathcal{A}^*$  avec  $\mathcal{A}$  un alphabet muni d'un ordre total  $<_{\mathcal{A}}$ .

L'évaluation de  $w$ , notée  $\text{ev}(w)$ , est le vecteur  $(a_1^{j_1}, a_2^{j_2}, \dots, a_k^{j_k})$  où chaque symbole  $a_i$ , ordonné suivant l'ordre  $<_{\mathcal{A}}$ , correspond un symbole distinct de  $w$ , et l'exposant  $j_i$  correspond au nombre d'occurrences du symbole  $a_i$ .

**Exemple 12.2.2 :** Sur l'alphabet des entiers ordonnés naturellement, on a l'évaluation suivante :

$$\text{ev}(98723138814) = (1^2, 2^1, 3^2, 4^1, 7^1, 8^3, 9^1).$$

Soit  $\chi : \mathbb{N} \rightarrow \mathbb{N}$  une fonction croissante telle que  $\chi(0) = 0$ . Soit  $w$  un mot d'évaluation  $(a_1^{j_1}, \dots, a_k^{j_k})$ . On pose

$$D_{\chi}(w, a_i, a_j) := \chi(1 + |w|_{< a_i}) - \chi(1 + |w|_{< a_j}),$$

avec  $|w|_{< a_k}$  le nombre de symboles de  $w$  strictement plus petit que  $a_k$ .

**Exemple 12.2.3 :** Soit  $w := 4292481$  un mot d'évaluation  $(1^1, 2^2, 4^2, 8^1, 9^1)$  et soit  $\chi : m \mapsto m^2$ .

$$\begin{aligned} D_{\chi}(w, 9, 4) &= \chi(1 + |w|_{< 9}) - \chi(1 + |w|_{< 4}), \\ &= \chi(7) - \chi(4) = 49 - 16 = 33. \end{aligned}$$

À partir de l'évaluation et de la fonction de différence  $D_{\chi}$ , on introduit la notion de *points de rupture*. Ces points nous seront utiles pour associer à chaque mot une  $\chi$ -fonction de parking. Ces points de rupture vont être associés aux abscisses des marches qui collent à la courbe discrète  $\chi$  dans la  $\chi$ -fonction de parking (cf. § 3.3.4 et § 4.1.1).

**Définition 12.2.4 :** Un symbole  $a_i$  est un  $\chi$ -point de rupture de  $w$  si

$$a_i = \min(w) \quad \text{ou} \quad D_{\mathcal{A}}(a_i, a_j) \geq D_{\chi}(w, a_i, a_j), \quad \text{pour tout symbole } a_j <_{\mathcal{A}} a_i.$$

On note  $\text{PR}_{\chi}(w)$  l'ensemble des  $\chi$ -points de rupture de  $w$ . On dit que  $P(a)$  est le  $\chi$ -point de rupture associé au symbole  $a$  de  $w$  si  $P(a)$  est le plus grand point de rupture de  $w$  plus petit que  $a$ .

**Exemple 12.2.5 :** L'ensemble des Id-points de rupture de  $w := 8312391$  est

$$\text{PR}_{\text{Id}}(w) = \{1, 3, 8, 9\}.$$

Le symbole 1 est le point de rupture associé à 2.

On définit (récursivement) l'application :

$$s_w : \begin{cases} \mathcal{A} & \longrightarrow \mathbb{N} \\ a_i & \longmapsto \begin{cases} \chi(1 + |w|_{<a_i}) & \text{si } a_i \text{ est un point de rupture,} \\ s_w(P(a_i)) + D_{\mathcal{A}}(a_i, P(a_i)) & \text{sinon.} \end{cases} \end{cases} \quad (12.2)$$

À partir de ce dernier outil, on définit alors la parkisation comme la donnée qui associe, à tout alphabet  $(\mathcal{A}, D_{\mathcal{A}})$ , l'application  $\text{park}_{\mathcal{A}}$  :

$$\text{park}_{\mathcal{A}}(w) := (Q_i), \quad \text{où } Q_i := \{j \mid s_w(a_j) = i\}. \quad (12.3)$$

pour tout mot  $w$  de  $\mathcal{A}^*$ .

**Remarque 12.2.6 :** Si on exprime les fonctions de parking comme des mots (§ 3.3.3), l'application  $\text{park}_{\mathcal{A}}$  se définit alors par : soit  $w := a_1 a_2 \cdots a_m$ ,

$$\text{park}_{\mathcal{A}}(w) = s_w(a_1) s_w(a_2) \cdots s_w(a_m).$$

**Proposition 12.2.7 :**  $\text{park}_{\mathcal{A}}(w)$  est une  $\chi$ -fonction de parking, pour tout mot  $w \in \mathcal{A}^*$ .

PREUVE: Soit  $w \in \mathcal{A}^*$  un mot d'évaluation  $(a_1^{j_1}, a_2^{j_2}, \dots, a_k^{j_k})$ .

Par induction sur l'évaluation de  $w$ , on montre que  $\text{park}_{\mathcal{A}}(w) = (Q_i)$  satisfait la contrainte de parking (4.1).

Par définition de  $s_w$ , si  $a_h$  est un point de rupture, son image est définie en fonction du nombre de symboles strictement plus petits que  $a_h$ , c'est-à-dire en fonction de

$$|w|_{<a_h} = \sum_{i=1}^{h-1} j_i.$$

La parkisation associe alors les  $j_h$  positions du symbole  $a_h$  à l'ensemble  $Q_c$  avec  $c := \chi(1 + |w|_{<a_h})$ . Si les  $|w|_{<a_h}$  positions des symboles strictement plus petits que  $a_h$  sont réparties dans des ensembles d'indices strictement plus petits que  $c$  alors on a :

$$\sum_{i=1}^c \#Q_i \geq 1 + |w|_{<a_h}.$$

Si  $a_h$  n'est pas un point de rupture alors ces positions dans  $w$  sont associées à  $Q_d$  avec  $d$  défini en fonction du point de rupture associé  $p := P(a_h)$  par :

$$d := s_w(a_h) = s_w(p) + D_{\mathcal{A}}(a_h, p),$$

comme  $p$  est un point de rupture et  $a_h$  non, on déduit l'inégalité :

$$d < \chi(1 + |w|_{<p}) + D_\chi(w, a_h, p),$$

par définition de  $D_\chi$  :

$$\begin{aligned} \chi(1 + |w|_{<p}) + D_\chi(w, a_h, p) &= \chi(1 + |w|_{<p}) + \chi(1 + |w|_{<a_h}) - \chi(1 + |w|_{<p}) , \\ &= \chi(1 + |w|_{<a_h}) . \end{aligned}$$

On obtient donc  $\chi(1 + |w|_{<a_h}) > d$ . Ainsi si les positions des symboles strictement plus petits que  $a_h$  sont répartis dans des ensembles d'indices strictement plus petits que  $d$ , alors la contrainte suivante est vérifiée :

$$\begin{aligned} \sum_{i=1}^{\chi(1+|w|_{<a_h})} \#Q_i &\geq \sum_{i=1}^d \#Q_i = |w|_{\leq a_h} := |w|_{<a_h} + j_h , \\ &\geq 1 + |w|_{<a_h} . \end{aligned}$$

Reste à montrer que la fondation du raisonnement est solide. Par définition, le symbole minimal  $a_1$  est un point de rupture, donc ces positions dans  $w$  sont réparties dans  $Q_{\chi(1)}$ . On a bien

$$\sum_{i=1}^{\chi(1)} \#Q_i \geq 1 ,$$

et ainsi par induction, on a montré que la contrainte de parking (4.1) est satisfaite :

$$\sum_{i=1}^{\chi(h)} \#Q_i \geq h , \quad \text{pour tout } h \leq |w| .$$

Les images de  $\text{park}_{\mathcal{A}}$  sont donc des  $\chi$ -fonctions de parking. ■

### 12.2.2. Application partitionnante

#### Surjectivité, graduation et commutativité

Tout d'abord, la définition de  $\text{park}_{\mathcal{A}}$  (12.3) associe à tout mot de longueur  $n$  une séquence d'ensembles disjoints qui couvrent  $[n]$ . On en déduit que  $\text{park}_{\mathcal{A}}$  est une application graduée (Condition 2).

À partir de la définition (12.2), quelque soit le mot  $w$  (non-vide), le plus petit indice d'ensemble non-vide de la  $\chi$ -fonction de parking  $\text{park}_{\mathcal{A}}(w)$  est nécessairement  $\chi(1)$ . D'où si  $\chi(1) > 1$  alors nécessairement  $\text{park}_{\mathcal{A}}$  n'est pas surjective.

On suppose alors  $\chi(1) = 1$  et on considère les  $\chi$ -fonctions de parking en terme de mots (cf. § 3.3.3) :

$$w_1 \cdots w_n \quad \rightsquigarrow \quad (Q_i)_{i \in [\chi(n)]} \quad \text{avec} \quad Q_i = \{j \in [n] \mid w_j = i\}. \quad (2.1')$$

Soit  $w$  une  $\chi$ -fonction de parking, définie comme un mot d'évaluation  $(a_1^{j_1}, \dots, a_k^{j_k})$ .

À partir de la Remarque 12.2.6, on déduit que  $D_\chi(w, a_c, a_d) = D_{\mathbb{N}}(a_c, a_d)$ , pour tout  $c, d \in [k]$ . Ceci implique que  $s_w(a_i) = a_i$ , pour tout  $i \in [k]$ ; et enfin que  $\text{park}_{\mathbb{N}}(w) = w$ . D'où si  $\chi(1) = 1$  alors  $\text{park}_{\mathbb{N}}$  est surjective.

Par ailleurs, pour tout  $n$  et tout alphabet  $(\mathcal{A}, D_{\mathcal{A}})$ , on peut définir une injection  $I$  de  $[\chi(n)]$  dans  $\mathcal{A}$  qui préserve les différences : (c'est-à-dire  $i - j = D_{\mathcal{A}}(I(i), I(j))$ , pour tout  $i, j \in [\chi(n)]$ ). On déduit alors que  $s_{I(w)}(I(i)) = s_w(i)$  et ainsi que  $\text{park}_{\mathcal{A}}$  est aussi surjective. Autrement dit, cela signifie :

**Lemme 12.2.8 (Condition 1) :** *L'application  $\text{park}_{\mathcal{A}}$  est surjective si et seulement si  $\chi(1) = 1$ .*

Les applications gauche et droite sont des morphismes injectifs par définition du type d'alphabet. En conséquence de la preuve de ce lemme, on a alors :

**Corollaire 12.2.9 (Condition 3) :** *L'application  $\text{park}_{\mathcal{A}}$  commute avec les injections gauche  $g_{\mathcal{A}, \mathcal{B}}$  et droite  $d_{\mathcal{B}, \mathcal{A}}$  (quelque soit l'alphabet  $\mathcal{B}$ ).*

### Propriétés intermédiaires

Dans le précédent paragraphe, nous avons montré que les conditions de graduation et de commutativité avec les injections gauche et droite (Conditions 2 et 3) sont satisfaites quelque soit  $\chi$  et que la condition de surjectivité (Condition 1) n'est vérifiée que lorsque  $\chi(1) = 1$ .

Dans les deux prochains paragraphes, on distingue les cas où  $\chi$  est assimilée à une courbe convexe ou à une courbe concave et on montre que les deux dernières conditions (Conditions 4 et 5) sont satisfaites si et seulement si  $\chi$  est une droite.

En avant-propos à ces paragraphes, on effectue un petit rappel : une fonction discrète  $f$  est *convexe* (sur  $\mathbb{N}$ ) si

$$f(i) - f(i-1) \leq f(i+1) - f(i), \quad \text{pour tout entier } i,$$

et elle est *concave* si

$$f(i) - f(i-1) \geq f(i+1) - f(i), \quad \text{pour tout entier } i.$$

Voici quelques caractéristiques de la parkisation :

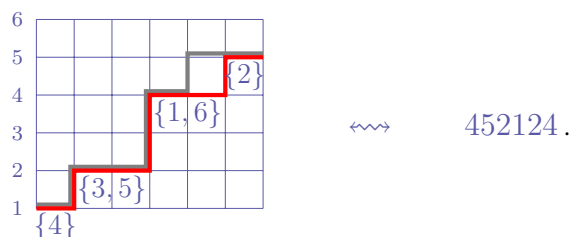
**Remarque 12.2.10 :** Dans ce paragraphe et les suivants, on considère les fonctions de parking comme des mots  $f := f_1 f_2 \cdots f_n$ .

Soit  $w := a_1 a_2 \cdots a_n$  un mot de  $\mathcal{A}^*$  et son parkisé  $f := f_1 f_2 \cdots f_n = \text{park}_{\mathcal{A}}(w)$ .

**Remarque 12.2.11 :** Par définition de  $s_w$ , si  $a_i = a_j$  alors  $f_i = f_j$ , pour toutes positions  $i$  et  $j$  de  $[n]$  (cf. Remarque 12.2.6).

La réciproque n'est pas nécessairement vrai. En effet, s'il existe un entier  $i$  tel que  $\chi(i) = \chi(i+1)$  alors cette réciproque est fausse :

**Exemple 12.2.12 :** Soit  $\chi$  l'application définie à partir des sommes partielles de la suite  $(1, 1, 0, 2, 1, 0, \dots)$ . On a  $\chi(2) = \chi(3)$  et  $\chi(5) = \chi(6)$ . Le mot  $w = 783157$  a pour parkisé 452124. On observe que la réciproque n'est pas vraie :  $s_w(3) = s_w(5) = 2$ .



On en conclut que :

**Proposition 12.2.13 :** La longueur du vecteur d'évaluation de  $w$  est supérieure (ou égale) à celle du vecteur d'évaluation de  $f$ . Elle est égale si et seulement si  $\chi(i) < \chi(i+1)$ , pour tout entier  $i > 0$ .

PREUVE : Par définition de  $s_w$  (12.2), deux symboles ne peuvent avoir la même valeur que s'ils sont des points de rupture (cf. Exemple 12.2.12). Si un symbole n'est pas un point de rupture alors sa valeur dépend de sa différence avec son point de rupture associé ; et par définition du type des alphabets avec différences, deux symboles ne peuvent avoir la même différence un même troisième que s'ils sont égaux.

Par ailleurs, si  $\chi(i) < \chi(i+1)$ , pour tout entier  $i > 0$ , alors chaque symbole (de l'évaluation) a alors une image distincte par  $s_w$ . D'après la Remarque 12.2.11, chaque occurrence d'un même symbole a même image d'où  $\ell(\text{ev}(w)) = \ell(\text{ev}(f))$ . ■

On a vu, dans le précédent paragraphe, que si on considère les  $\chi$ -fonctions de parking comme des mots alors on a  $\text{park}_{\mathbb{N}}(f) = f$ . On peut donc extraire une notion de points de rupture analogue sur les  $\chi$ -fonctions de parking  $f := f_1 f_2 \cdots f_n$  :

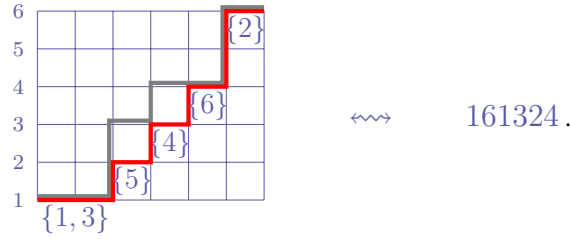
$$\text{PR}_{\chi}(f) = \left\{ f_j \mid j \in [n] \text{ et } \chi(1 + |f|_{<f_j}) = f_j \right\}. \quad (12.4)$$

**Remarque 12.2.14 :** La définition des points de rupture d'une fonction de parking se déduit de la définition de  $s_w$ . Si  $a$  est un point de rupture  $s_w(a) = \chi(j)$  avec  $j = 1 + |w|_{<a}$ , et sinon  $s_w(a) < \chi(j)$ .

**Remarque 12.2.15 :** Cette définition des points de rupture d'une  $\chi$ -fonction de parking ne coïncide avec la Définition 12.2.4 que lorsque  $\chi(1) = 1$ . Dans la mesure où l'on souhaite caractériser **park** comme une application partitionnante, les autres cas, avec  $\chi(1) > 1$ , n'ont aucun intérêt pour la suite (Lemme 12.2.8), on s'autorise donc cette définition concise (12.4).

Ces points de rupture se traduisent sur les fonctions de parking, en tant que chemins décorés (§ 3.3.4 et Chapitre 4), comme l'ensemble des abscisses où les marches du chemin décoré coïncident avec la courbe discrète  $\chi$ .

**Exemple 12.2.16 :** Dans la  $\chi$ -fonction de parking (où  $\chi$  est définie à partir des sommes partielles de la suite  $(1, 0, 2, 1, 0, 2, \dots)$ ) :



les points de rupture sont (les abscisses ou symboles) 1, 4 et 6 respectivement associés aux ensembles (de positions)  $\{1, 3\}$ ,  $\{6\}$  et  $\{2\}$ .

Dans l'exemple précédent, tous les indices d'ensembles non-vides sont des points de rupture.

Avec analogie avec les mots, on note  $P(f_k)$  le point de rupture associé à  $f_k$ . Ce dernier est défini comme le plus grand point de rupture de la fonction de parking plus petit que  $f_k$ .

Soit  $w = a_1 a_2 \dots a_n$  un mot de  $\mathcal{A}^*$  et soit  $f = f_1 f_2 \dots f_n$  son parkisé.

**Proposition 12.2.17 :** Les propriétés suivantes sont satisfaites :

- $a_k \in \text{PR}_\chi(w)$  si et seulement si  $f_k \in \text{PR}_\chi(f)$ ,
  - si  $a_k \notin \text{PR}_\chi(w)$  alors  $D_{\mathcal{A}}(a_k, P(a_k)) = f_k - P(f_k)$ ,
- pour tout  $k \in [n]$ .

PREUVE :

- si  $a_k \in \text{PR}_\chi(w)$  alors  $f_k = s_w(a_k) = \chi(1 + |w|_{<a_k})$  d'où  $f_k$  est un point de rupture. Réciproquement par contraposée, si  $a_k \notin \text{PR}_\chi(w)$ ,

$$s_w(a_k) = s_w(P(a_k)) + D_{\mathcal{A}}(a_k, P(a_k)), \quad (12.5)$$

par définition de  $s_w$  et des points de rupture (Définition 12.2.4) :

$$\begin{aligned} &< \chi(1 + |w|_{<P(a_k)}) + \chi(1 + |w|_{<a_k}) - \chi(1 + |w|_{<P(a_k)}), \\ &= \chi(1 + |w|_{<a_k}) \end{aligned}$$

Donc si  $a_k$  n'est pas un point de rupture alors  $s_w(a_k)$  n'est pas un point de rupture.

- Immédiatement à partir de (12.5). ■

## Compatibilité aux facteurs

À partir des propriétés précédentes, on étudie dans quels cas la compatibilité aux facteurs est satisfaite :

$$\text{si } \text{park}_{\mathcal{A}}(uv) = \text{park}_{\mathcal{A}}(u'v') \quad \text{alors} \quad \begin{cases} \text{park}_{\mathcal{A}}(u) = \text{park}_{\mathcal{A}}(u'), \\ \text{park}_{\mathcal{A}}(v) = \text{park}_{\mathcal{A}}(v'), \end{cases} \quad [\text{Condition 4}]$$

avec  $u, u' \in \mathcal{A}^m$  et  $v, v' \in \mathcal{A}^n$ .

**Proposition 12.2.18 :** Pour tout mot  $w \in \mathcal{A}^*$  et tout facteur  $u$  de  $w$ , si  $\chi$  est convexe, pour tout  $i > 1$  alors  $\text{PR}_{\chi}(w) \cap \mathcal{A}(u) \subset \text{PR}_{\chi}(u)$ .

PREUVE : On suppose  $\chi$  convexe. Soit  $w$  un mot de  $\mathcal{A}^*$  et soit  $u$  un facteur de  $w$ . On suppose  $a$  un point de rupture de  $w$  présent dans  $u$  (autrement dit  $a$  est un symbole de  $\text{PR}_{\chi}(w) \cap \mathcal{A}(u)$ ). Par définition des points de rupture, si  $a \in \text{PR}_{\chi}(w)$  alors pour tout symbole  $b$  tel que  $b <_{\mathcal{A}} a$ ,

$$D_{\mathcal{A}}(a, b) \geq D_{\chi}(w, a, b) = \chi(1 + |w|_{<a}) - \chi(1 + |w|_{<b})$$

Comme  $u$  est un facteur de  $w$ , on a l'identité :

$$|w|_{<a} - |w|_{<b} = \alpha + |u|_{<a} - |u|_{<b},$$

avec  $\alpha$  le nombre de symboles de  $w$  strictement plus petit que  $a$  et plus grand que  $b$  dans le sous-mot complémentaire du facteur  $u$  dans  $w$ . Comme  $\alpha \geq 0$ , on a alors,

$$|w|_{<a} - |w|_{<b} \geq |u|_{<a} - |u|_{<b}.$$

Par hypothèse,  $\chi$  est une fonction croissante convexe, on a alors

$$\chi(1 + |w|_{<a}) - \chi(1 + |w|_{<b}) \geq \chi(1 + |u|_{<a}) - \chi(1 + |u|_{<b}).$$

On a donc l'inégalité,

$$D_{\mathcal{A}}(a, b) \geq D_{\chi}(w, a, b) \geq D_{\chi}(u, a, b).$$

Le symbole  $a$  est donc un point de rupture de  $u$ . ■

À partir de cette proposition, on déduit :

**Lemme 12.2.19 (Condition 4) :** La parkisation satisfait la compatibilité aux facteurs si et seulement si  $\chi$  est convexe, pour tout entier  $i > 1$ .

PREUVE : Soient  $w$  et  $w'$  deux mots de  $\mathcal{A}^n$  tels que  $\text{park}_{\mathcal{A}}(w) = \text{park}_{\mathcal{A}}(w')$ .

Par définition, si un symbole  $a$  n'est pas un point de rupture, alors  $s_w(a)$  est défini en fonction de son point de rupture associé :  $s_w(P(a)) + D_{\mathcal{A}}(a, P(a))$ .

À partir de la Proposition 12.2.17, on sait que si  $a$  est un point de rupture de  $u$  alors  $s_w(a)$  est un point de rupture pour la fonction de parking. Si ce n'est pas le cas, la différence avec son point de rupture associé est égal à la différence des images par  $s_w$ . D'où si  $\text{park}_{\mathcal{A}}(w) = \text{park}_{\mathcal{A}}(w')$  alors pour tous symboles  $a$  dans  $w$  et  $b$  dans  $w'$  qui ne sont pas des points de rupture :



si  $s_w(a) = s_{w'}(b)$  alors  $D_{\mathcal{A}}(a, P(a)) = D_{\mathcal{B}}(b, P(b))$ .

À partir de la Proposition 12.2.18, on déduit alors que si  $\chi$  est convexe, pour tout  $i > 1$  alors  $\text{park}_{\mathcal{A}}(w[j, k]) = \text{park}_{\mathcal{A}}(w'[j, k])$ , pour tout  $1 \leq j \leq k \leq |w|$ .

Réciproquement par contraposée : On suppose que l'application  $\chi$  n'est pas convexe. Soit  $(x_1, x_2, \dots)$  le vecteur d'entiers où  $x_1 = \chi(1)$  et  $x_i = \chi(i) - \chi(i-1)$ , pour tout  $i > 1$ . Si  $\chi$  n'est pas convexe, il existe un entier  $i$  tel que  $x_{i-1} > x_i$ . On suppose  $i$  le plus petit entier tel que  $x_{i-1} > x_i$ .

On montre cette contraposée en exhibant un contreexemple générique sur l'alphabet des entiers (où  $D_{\mathbb{N}}(a, b) := a - b$ , cf. Exemple 12.2.20). On pose  $w$  et  $w'$  les mots croissants de longueur  $i$  définis par :

$$w = \chi(1)\chi(2) \cdots \chi(i) \quad \text{et} \quad w' = \chi(1)\chi(2) \cdots (\chi(i) + 1).$$

Par construction, tous les symboles de  $w$  et  $w'$  sont des points de rupture, et en particulier, on a

$$\begin{aligned} D_{\mathbb{N}}(\chi(i), \chi(i-1)) &= D_{\mathbb{N}}(\chi(i) + 1, \chi(i-1)) - 1, \\ &= D_{\chi}(w, \chi(i), \chi(i-1)) = D_{\chi}(w', \chi(i) + 1, \chi(i-1)), \\ &= \chi(1 + |w|_{<\chi(i)}) - \chi(1 + |w|_{<\chi(i-1)}) = \chi(1 + |w'|_{<\chi(i)+1}) - \chi(1 + |w'|_{<\chi(i-1)}), \\ &= \chi(i) - \chi(i-1) = x_i. \end{aligned}$$

Le parkisé de ces deux mots est bien le même :

$$\text{park}_{\mathbb{N}}(w) = \text{park}_{\mathbb{N}}(w') = w.$$

On considère les suffixes  $u := w[2, i]$  et  $u' = w'[2, i]$ . On a

$$\begin{aligned} D_{\chi}(u, \chi(i), \chi(i-1)) &= \chi(1 + |u|_{<\chi(i)}) - \chi(1 + |u|_{<\chi(i-1)}), \\ &= \chi(i-1) - \chi(i-2) = x_{i-1}. \end{aligned}$$

Par hypothèse  $x_i < x_{i-1}$ , donc

$$D_{\mathbb{N}}(\chi(i), \chi(i-1)) < D_{\chi}(u, \chi(i), \chi(i-1)).$$

Autrement dit  $\chi(i)$  n'est pas un point de rupture de  $u$ . Comme  $i$  est le plus petit indice pour lequel  $\chi$  n'est pas convexe, à partir de la Proposition 12.2.18, on déduit que, excepté le dernier symbole, tous les symboles de  $u$  sont des points de rupture, d'où

$$\text{park}_{\mathbb{N}}(u) = \chi(1)\chi(2) \cdots \chi(i-2)(\chi(i-2) + x_i).$$

De la même manière avec  $u'$ , on déduit que

$$D_{\mathbb{N}}(\chi(i) + 1, \chi(i-1)) = x_i + 1 \leq x_{i-1} = D_{\chi}(u', \chi(i) + 1, \chi(i-1)).$$

Le parkisé de  $u'$  est alors :

$$\text{park}_{\mathbb{N}}(u') = \chi(1)\chi(2) \cdots \chi(i-2)(\chi(i-2) + x_i + 1).$$

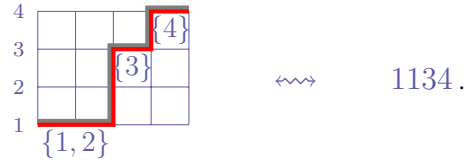
(Remarque : si  $\chi(i)$  est un point de rupture de  $u'$ , cela signifie que  $x_i + 1 = x_{i-1}$  alors

$$s_w(\chi(i)) = \chi(i-2) + x_{i-1} = \chi(i-1).)$$

D'où  $u$  et  $u'$  n'ont pas le même parkisé et la condition de compatibilité aux facteurs n'est pas satisfaite.

Ce contreexemple générique s'étend naturellement à tout alphabet  $\mathcal{A}$ , par morphisme injectif (d'alphabet avec différences). D'où si  $\chi$  est convexe alors la compatibilité aux facteurs est satisfaite. ■

**Exemple 12.2.20 :** Soit  $\chi$  l'application définie à partir des sommes partielles de la suite  $(1, 0, 2, 1, 0, 2, \dots)$ . Les mots  $1134$  et  $1135$  ont pour parkisé la  $\chi$ -fonction de parking :



Les suffixes  $134$  et  $135$  ont respectivement les parkisés suivants :



Dans ce paragraphe et les deux précédents, nous avons trouvé sous quelles contraintes la parkisation satisfait les quatre premières conditions d'application partitionnante. Il nous reste à trouver sous quelles contraintes la parkisation satisfait la compatibilité au mélange.

### Compatibilité au mélange

Dans ce paragraphe, on étudie dans quels cas la compatibilité au mélange est satisfaite :

$$\text{si } \left\{ \begin{array}{l} \text{park}_{\mathcal{A}}(u) = \text{park}_{\mathcal{A}}(u') \\ \text{park}_{\mathcal{B}}(v) = \text{park}_{\mathcal{B}}(v') \end{array} \right\} \text{ alors } \text{park}_{\mathcal{A} \oplus \mathcal{B}}((uv) \cdot \sigma) = \text{park}_{\mathcal{A} \oplus \mathcal{B}}((u'v') \cdot \sigma), \quad (5')$$

pour tous mots  $u, u' \in \mathcal{A}^m$ , tous mots  $v, v' \in \mathcal{B}^n$  et toute permutation  $\sigma \in \text{Sh}_{m,n}$ .

Comme l'application  $s_w$  est définie indépendamment des positions des symboles de  $w$ , on commence par déduire que :

**Proposition 12.2.21 :** La parkisation est équivariante pour l'action du groupe symétrique :

$$\text{park}_{\mathcal{A}}(w \cdot \sigma) = \text{park}_{\mathcal{A}}(w) \cdot \sigma ,$$

pour tout mot  $w$  et toute permutation  $\sigma$ .

Il suffit alors de caractériser sous quelles contraintes, on satisfait

$$\text{si } \left\{ \begin{array}{l} \text{park}_{\mathcal{A}}(u) = \text{park}_{\mathcal{A}}(u') \\ \text{park}_{\mathcal{B}}(v) = \text{park}_{\mathcal{B}}(v') \end{array} \right\} \text{ alors } \text{park}_{\mathcal{A} \oplus \mathcal{B}}(uv) = \text{park}_{\mathcal{A} \oplus \mathcal{B}}(u'v') . \quad (5'')$$

Soient  $u \in \mathcal{A}^*$  et  $v \in \mathcal{B}^*$  deux mots soit  $uv \in (\mathcal{A} \oplus \mathcal{B})^*$  la concaténation  $g_{\mathcal{A}, \mathcal{B}}(u)$  et  $d_{\mathcal{A}, \mathcal{B}}(v)$ .

**Proposition 12.2.22 :** Si  $\chi$  est concave, pour tout  $i > 1$ , alors  $\text{PR}_{\chi}(u) \cup \text{PR}_{\chi}(v) \subset \text{PR}_{\chi}(uv)$ .

PREUVE: Par définition du type des alphabets avec différence, on a  $D_{\mathcal{A} \oplus \mathcal{B}}(b, a) = +\infty$ , pour tout  $a \in \mathcal{A}$  et tout  $b \in \mathcal{B}$ . On déduit immédiatement de la Définition 12.2.4, des points de rupture, que l'ensemble des points de rupture de  $u$  est donc nécessairement inclus dans celui de  $uv$ , indépendamment de  $\chi$ . Il en va de même pour le minimum  $b$  de  $v$ , il vérifie bien

$$D_{\mathcal{A} \oplus \mathcal{B}}(b, a) = \infty > \chi(1 + |u|) - \chi(1 + |u|_{<a}) ,$$

pour tout symbole  $a$  de  $u$  (nécessairement plus petit que  $b$ ).

Pour les autres points de rupture  $b$  de  $v$ , on souhaite vérifier à quelles conditions, on a

$$D_{\mathcal{A} \oplus \mathcal{B}}(b, a) \geq \chi(1 + |uv|_{<b}) - \chi(1 + |uv|_{<a}) .$$

Quelque soit le symbole  $a$  s'il est dans  $u$  alors la contrainte est satisfaite. Si  $a$  est dans  $v$ , on a :

$$\chi(1 + |uv|_{<b}) - \chi(1 + |uv|_{<a}) = \chi(1 + |u| + |v|_{<b}) - \chi(1 + |u| + |v|_{<a}) .$$

Si  $\chi$  est concave, on a alors

$$\chi(1 + |v|_{<b}) - \chi(1 + |v|_{<a}) \geq \chi(1 + |u| + |v|_{<b}) - \chi(1 + |u| + |v|_{<a}) .$$

D'où si  $b \in \text{PR}_{\chi}(u)$  :

$$D_{\mathcal{A} \oplus \mathcal{B}}(b, a) \geq \chi(1 + |u| + |v|_{<b}) - \chi(1 + |u| + |v|_{<a}) .$$

On a donc obtenu que si  $\chi$  est concave et  $b \in \text{PR}_{\chi}(v)$  alors  $b \in \text{PR}_{\chi}(uv)$  et d'où si  $\chi$  est concave alors

$$\text{PR}_{\chi}(u) \cup \text{PR}_{\chi}(v) \subset \text{PR}_{\chi}(uv) . \quad \blacksquare$$

Avec analogie avec le précédent paragraphe, à partir de cette proposition, on déduit que

**Lemme 12.2.23 (Condition 5):** *La parkisation satisfait la compatibilité au mélange si et seulement si  $\chi$  est concave, pour tout entier  $i > 1$ .*

PREUVE : Soient  $u, u' \in \mathcal{A}^*$  et  $v, v' \in \mathcal{B}^*$  tels que

$$\text{park}_{\mathcal{A}}(u) = \text{park}_{\mathcal{A}}(u') \quad \text{et} \quad \text{park}_{\mathcal{B}}(v) = \text{park}_{\mathcal{B}}(v').$$

On suppose  $\chi$  concave pour tout entier  $i > 1$ . À partir de la Proposition 12.2.17, on sait que, pour tout mot  $w$ , si  $a$  est un point de rupture de  $u$  alors  $s_w(a)$  est un point de rupture pour la fonction de parking. Si ce n'est pas le cas, la différence avec son point de rupture associé est égale à la différence des images par  $s_w$ . D'où pour tous symboles  $a$  dans  $u$  (resp.  $v$ ) et  $b$  dans  $u'$  (resp.  $v'$ ) qui ne sont pas des points de rupture :

$$\text{si } s_u(a) = s_{u'}(b) \text{ alors } D_{\mathcal{A}}(a, P(a)) = D_{\mathcal{A}}(b, P(b)),$$

(et respectivement si  $s_v(a) = s_{v'}(b)$  alors  $D_{\mathcal{B}}(a, P(a)) = D_{\mathcal{B}}(b, P(b))$ ).

Ainsi de la Proposition 12.2.22, on déduit que si  $\chi$  est concave, pour tout  $i > 1$ , alors  $\text{park}_{\mathcal{A} \oplus \mathcal{B}}(uv) = \text{park}_{\mathcal{A} \oplus \mathcal{B}}(u'v')$ .

Réciproquement, on montre par contraposée que si  $\chi$  n'est pas concave alors il existe des mots  $v$  pour lesquels certains points de rupture  $b$  n'appartiennent pas à l'ensemble des points de rupture de  $uv$ . De manière similaire au Lemme 12.2.19, on construit un contreexemple générique à partir d'alphabets d'entiers (cf. Exemple 12.2.24) et ce dernier s'étend à tout alphabet par morphisme injectif.

Soit  $(x_1, x_2, \dots)$  le vecteur défini par  $x_1 = \chi(1)$  et  $x_i = \chi(i) - \chi(i-1)$ , pour tout  $i > 1$ . Si  $\chi$  n'est pas concave, il existe un entier  $i$  tel que  $x_i < x_{i+1}$ . On suppose  $i$  le plus petit de ces entiers. On considère  $\mathbb{N}$  et  $\mathbb{N}'$  deux alphabets d'entiers. On décore d'un ' les entiers de  $\mathbb{N}'$  (par exemple :  $1', 2', 3', \dots$ ). On pose  $u \in \mathbb{N}^*$  et  $v, v' \in \mathbb{N}'^*$  les mots croissants :

$$u = \chi(1) \quad \text{et} \quad \begin{cases} v = \chi(1)' \cdots \chi(i-1)'\chi(i)', \\ v' = \chi(1)' \cdots \chi(i-1)'(\chi(i) + 1)'. \end{cases}$$

Chaque symbole de chaque mot est un point de rupture, et en particulier :

$$\begin{aligned} D_{\mathbb{N}'}((\chi(i) + 1)', \chi(i-1)') - 1 &= D_{\chi}(v', \chi(i) + 1)', \chi(i-1)'), \\ &= D_{\mathbb{N}'}(\chi(i)', \chi(i-1)'), \\ &= x_i. \end{aligned}$$

Les mots  $v$  et  $v'$  ont le même parkisé :

$$\text{park}_{\mathbb{N}'}(v) = \text{park}_{\mathbb{N}'}(v') = \chi(1) \cdots \chi(i-1)\chi(i).$$

Cela étant, comme  $x_i < x_{i+1}$ , les concaténations  $uv$  et  $uv'$  n'ont pas le même parkisé :

$$\begin{aligned}\text{park}_{\mathbb{N} \oplus \mathbb{N}'}(uv) &= \chi(1) \cdots \chi(i)(\chi(i) + x_i), \\ \text{park}_{\mathbb{N} \oplus \mathbb{N}'}(uv') &= \chi(1) \cdots \chi(i)(\chi(i) + x_i + 1).\end{aligned}$$

Donc la parkisation est compatible avec le mélange si et seulement si  $\chi$  est concave. ■

**Exemple 12.2.24 :** Soit  $\chi$  définie à partir des sommes partielles du vecteur  $(1, 1, 2, 1, \dots)$ . Soit  $u := 1$  un mot de  $\mathbb{N}^*$  et soient  $v := 12$  et  $v' := 13$  deux mots de  $\mathbb{N}^*$ . Les mots  $u$ ,  $v$  et  $v'$  ont pour parkisé les  $\chi$ -fonctions de parking suivantes :

$$\text{park}_{\mathbb{N}^*}(u) = 1 \iff 1 \begin{array}{|c|} \hline \square \\ \hline \end{array} \begin{array}{|c|} \hline \textcolor{red}{\{1\}} \\ \hline \end{array}, \quad \text{et} \quad \text{park}_{\mathbb{N}^*}(v) = \text{park}_{\mathbb{N}^*}(v') = 12 \iff \begin{array}{|c|c|} \hline 2 & \square \\ \hline 1 & \begin{array}{|c|} \hline \textcolor{red}{\{2\}} \\ \hline \end{array} \\ \hline \end{array}.$$

Les concaténations  $uv$  et  $uv'$  sur la somme d'alphabet  $\mathbb{N} \oplus \mathbb{N}$  n'ont pas le même parkisé :

$$\text{park}_{\mathbb{N} \oplus \mathbb{N}}(uv) = 123 \iff \begin{array}{|c|c|c|} \hline 4 & & \square \\ \hline 3 & & \square \\ \hline 2 & \begin{array}{|c|} \hline \textcolor{red}{\{2\}} \\ \hline \end{array} & \begin{array}{|c|} \hline \textcolor{red}{\{3\}} \\ \hline \end{array} \\ \hline 1 & \begin{array}{|c|} \hline \textcolor{red}{\{1\}} \\ \hline \end{array} & \square \\ \hline \end{array} \quad \text{et} \quad \text{park}_{\mathbb{N} \oplus \mathbb{N}}(uv') = 124 \iff \begin{array}{|c|c|c|} \hline 4 & & \begin{array}{|c|} \hline \textcolor{red}{\{3\}} \\ \hline \end{array} \\ \hline 3 & & \square \\ \hline 2 & \begin{array}{|c|} \hline \textcolor{red}{\{2\}} \\ \hline \end{array} & \square \\ \hline 1 & \begin{array}{|c|} \hline \textcolor{red}{\{1\}} \\ \hline \end{array} & \square \\ \hline \end{array}.$$

### Application partitionnante

La Proposition 12.2.7 nous assure que la parkisation est bien une application qui associe, à chaque mot, une  $\chi$ -fonction de parking. À partir du Lemme 12.2.8, on sait que la parkisation est surjective (Condition 1) si et seulement si  $\chi(1) = 1$ . Comme l'application est par définition graduée, la Condition 2 de graduation est automatiquement satisfaite. Le Corollaire 12.2.9 nous assure que la parkisation commute avec les morphismes injectifs gauche et droit dans tous les cas (Condition 3). Les Lemmes 12.2.19 et 12.2.23 montrent que les compatibilités aux facteurs (Condition 4) et au mélange (Condition 5) sont vraies respectivement si et seulement si  $\chi$  est convexe, et si et seulement si  $\chi$  est concave. D'où

**Théorème 12.2.25 :** La parkisation est une application partitionnante si et seulement si  $\chi : n \mapsto 1 + m(n - 1)$ , avec  $m \geq 0$ .

# Perspectives

## Énumération d'automates

Le problème d'énumération des automates finis déterministes minimaux reste un problème ouvert dans le cas général. La technique utilisée précédemment (cf. [Pri15b] ou Partie II) apporte, certes, un nouvel angle d'attaque pour aborder le problème.

Mais dans le cas général, l'information n'est pas suffisante pour extraire une formule d'énumération des automates minimaux. Cela étant, ces techniques (pour le cas acyclique ou non) devraient permettre d'améliorer les outils de *tirage aléatoire* existants : échantillonneurs de Boltzmann [BDN08] ou générateurs par chaînes de Markov [CD12], etc..

La structure même des automates rend le problème bien plus difficile que dans le cas des automates acycliques. Contrairement à ces derniers, les automates quasi-distingués peuvent contenir deux composantes fortement connexes reconnaissant le même langage.

Le problème du traitement de ces composantes est résolu par l'algorithme de minimisation de Moore [HMU07]. L'idée générale de cet algorithme est de partitionner l'ensemble des états  $Q$  en construisant une séquence d'application :

$$\nu_i : Q \times \Sigma \longrightarrow \lambda_i$$

où  $\nu_i$  et  $\lambda_i$  satisfont les conditions suivantes :

$$\emptyset \neq \nu_i(q, a) \subset \nu_{i-1}(q, a), \quad \text{pour tout état } q \text{ et tout symbole } a \in \Sigma$$

et

$$\lambda_i := \bigcup_{p \in \lambda_{i-1}} p / \nu_{i-1}$$

pour tout  $i > 1$ . Pour  $i = 1$ ,  $\nu_1$  est sans contrainte et  $\lambda := \{A, Q \setminus A\}$ .

L'algorithme se termine lorsque  $\lambda_i = \lambda_{i-1}$ . Dans la dernière partition de la séquence, chaque part du partitionnement correspond à un ensemble d'états reconnaissant exactement un même unique langage. Lorsque cette partition est composée uniquement de singleton, l'automate de départ est distingué.

Comme l'algorithme est déterministe, l'ensemble de toutes les séquences  $(\lambda_i)$  telles que la dernière partition ne contient que des singletons correspond à l'ensemble des automates distingués.

Je conjecture que la série indicatrice de cycles de l'ensemble des automates non-initiaux s'exprime comme une somme sur l'ensemble de telles séquences. Par ailleurs,

cet algorithme fait apparaître différentes symétries pouvant se traiter en parallèle : sur l'ensemble d'états et l'ensemble de symboles de l'alphabet. Ces symétries doivent se résumer à de simples binomiaux et donc cette somme pourrait être optimisée.

## Algèbre Hopf : relation de type plaxique

Le formalisme des *applications partitionnantes* et des *réalisations polynomiales* d'algèbres de Hopf définit une fondation solide pour étudier les relations similaires aux relations plaxiques et les algèbres de Hopf comme celle de LODAY-RONCO (des arbres binaires, **PBT**) ou encore celle des tableaux de POIRIER-REUTENAEUR (des fonctions symétriques libres, **FSym**) [PR95]. Dans mon mémoire de master et premier papier [Pri13], j'ai caractérisé les bonnes propriétés des relations plaxiques ou relations de Donald E. KNUTh [Knu70] :

$$\begin{aligned} u \cdot \textcolor{blue}{a} \textcolor{red}{c} \textcolor{blue}{b} \cdot v &\sim_p u \cdot \textcolor{red}{c} \textcolor{blue}{a} \textcolor{blue}{b} \cdot v && \text{pour } \textcolor{red}{a} \leq \textcolor{blue}{b} < \textcolor{red}{c}, \\ u \cdot \textcolor{blue}{b} \textcolor{red}{a} \textcolor{blue}{c} \cdot v &\sim_p u \cdot \textcolor{blue}{b} \textcolor{red}{c} \textcolor{blue}{a} \cdot v && \text{pour } \textcolor{red}{a} < \textcolor{blue}{b} \leq \textcolor{red}{c}, \end{aligned}$$

permettant de quotienter **FQSym** pour obtenir automatiquement **FSym**.

Soit  $\phi$  une *application partitionnante* (cf. [HP15] ou Partie III) définie pour un certain type d'alphabet. Une relation  $\equiv_{\mathcal{A}}$  sur le monoïde libre  $\mathcal{A}^*$  est dite de type  $\phi$ -plaxique [Pri13] si :

- la relation  $\equiv_{\mathcal{A}}$  est une congruence,
- pour tout  $u, v \in \mathcal{A}^*$ , on a

$$u \equiv_{\mathcal{A}} v \text{ ssi } \begin{cases} \phi(u) \equiv_{\mathcal{A}} \phi(v) & \text{et,} \\ \text{ev}(u) = \text{ev}(v), \end{cases}$$

où  $\text{ev}(u)$  est l'évaluation du mot  $u$ .

- pour tout  $u, v \in (\mathcal{A} \oplus \mathcal{B})^*$ , on a

$$u \equiv_{(\mathcal{A} \oplus \mathcal{B})} v \implies \begin{cases} u|_{\mathcal{A}} \equiv_{\mathcal{A}} v|_{\mathcal{A}} & \text{et,} \\ u|_{\mathcal{B}} \equiv_{\mathcal{B}} v|_{\mathcal{B}}. \end{cases}$$

Le théorème principal de [Pri13, Théorème 1] se généralise à n'importe quelle algèbre de Hopf définie par une application partitionnante. À toute algèbre de Hopf  $\mathcal{H}_{\phi}(\mathcal{A})$  et à toute relation  $\equiv$  de type  $\phi$ -plaxique sur l'ensemble des mots de  $\mathcal{A}^*$ , le quotient  $\mathcal{H}_{\phi}(\mathcal{A}) / \equiv$  est un quotient de Hopf indexé par le quotient  $\mathcal{C} / \equiv$ .

En particulier, il devient naturel de considérer de nouvelles relations spécifiques aux fonctions de parking ; comme :

$$u \cdot \textcolor{blue}{b} \textcolor{red}{c} \cdot v \sim u \cdot \textcolor{red}{c} \textcolor{blue}{b} \cdot v \quad \text{avec } \ell_i \leq \textcolor{blue}{b}, \textcolor{red}{c} < \ell_j$$

où  $\ell_i$  et  $\ell_j$  sont deux points de rupture consécutifs de  $w$ . Ces relations spécifiques apportent toute une famille de quotient de **PQSym** qui n'ont pas d'analogue dans **FQSym**

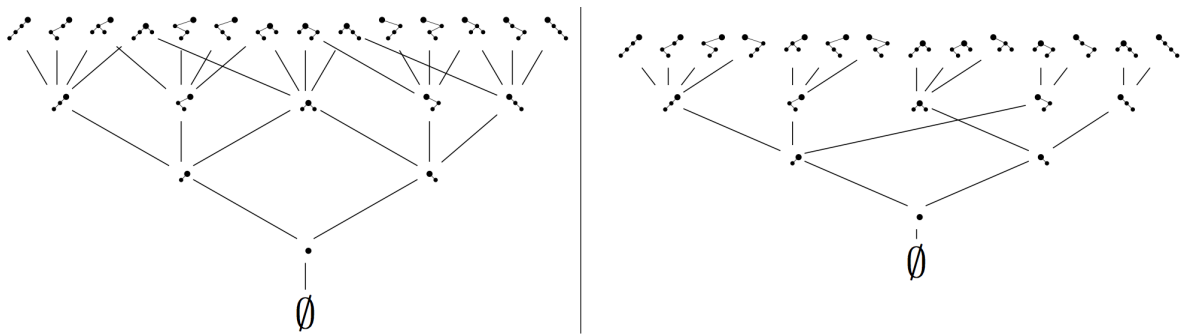
ou **WQSym**. Dans une autre direction, cette approche autorise à considérer des relations sur des alphabets munis de structures autres qu'une structure d'ordre. Dans le cas de l'algèbre de Hopf indexée par les antichaines couvrantes maximales, la réalisation polynomiale est définie sur un alphabet où les lettres sont des ensembles. La relation suivante :

$$u \cdot bvc \cdot w \sim u \cdot cvb \cdot w \quad \text{avec } b \cap c = \emptyset,$$

est alors de type plaxique.

Tout comme les classes plaxiques sont les fibres de l'algorithme de ROBINSON-SCHENSTED-KNUTH, on peut généralement associer à toute relation de type plaxique, un algorithme de type ROBINSON-SCHENSTED. Par exemple, la relation sylvestre [HNT02] est naturellement associée à l'algorithme d'insertion dans un arbre binaire de recherche.

Dans [Nze06], l'auteur avance un lien entre algèbres de Hopf et graphes en dualité de Sergey FOMIN [Fom94] :



Peut-on définir un lien générique entre les quotients de Hopf définis par des relations de type plaxique et des graphes en dualités de Fomin ? De même peut-on généraliser la notion de diagramme de croissance de Fomin pour des objets quotients des fonctions de parking ou des antichaines couvrantes ?

D'autre part, ces graphes en dualité apparaissent aussi indépendamment des algèbres de Hopf. Dans [Bet+14], les auteurs adaptent la description des diagrammes de croissance de Fomin pour manipuler des partitions planes. Les graphes en dualités sous-jacents peuvent-ils être associés à une algèbre de Hopf indexée par les partitions planes ?

God's Final Message to His Creation : We apologize for the inconvenience.  
(Douglas ADAMS, *So Long, and Thanks for All the Fish*)



# A. Développement Sage

SAGEMATH est un logiciel libre de mathématiques [Dev12]. Il combine la puissance de nombreux programmes libres ayant pour mission d'être une alternative viable libre et open source à MAGMA, MAPLE, MATHEMATICA et MATLAB.

Cet outil permet de faire des mathématiques de tout horizon : algèbre, analyse, théorie des nombres, cryptographie, analyse numérique, théorie des groupes, combinatoire, théorie des graphes, algèbre linéaire formelle, etc.

Ce logiciel est développé par et pour les chercheurs. Au cours de ma thèse, mes différents travaux de recherches ont largement utilisé l'exploration informatique avec SAGEMATH. Ma contribution à ce logiciel représente plusieurs milliers de lignes de code et documentation. Voici quelques-uns des tickets dont je suis l'auteur :

## Infrastructure algébrique : Sage's Categories

- #15573 (CO)PRODUCT COERCION : facilite la création et l'utilisation des algèbres de Hopf. Il fournit un code générique pour chercher automatiquement une base dans laquelle le produit ou le coproduit est défini (pas nécessairement la même) lorsque ces opérations ne sont pas définies dans la base utilisée.
- #15578 DEFAULT ANTIPODE : implémente la formule d'antipode des algèbres de Hopf graduées connexes de John W. MILNOR et John C. MOORE [MM65].
- #17367 CLASSES OF COMBINATORIAL STRUCTURES : formalise la notion de classe combinatoire dans une catégorie SAGE. Ce ticket propose aussi un framework épuré et simple pour implémenter facilement une classe combinatoire (sans se soucier de l'utilisation des `ClasscallMetaclass` par exemple).
- #18350 ADAMS OPERATOR : implémente les opérateurs d'Adams des algèbres de Hopf [AL15] (c'est-à-dire les produits de convolution itérés appliqués à l'identité).
- #18675 ADD 'CONNECTED' AS A CLASS FOR GRADED HOPF ALGEBRAS WITH BASIS : ajoute l'axiome connexe à la catégorie des algèbres de Hopf.
- #19282 OPERATORS EXPANDER IN CATEGORIES : est un outil pour faciliter la création de catégories intégrant des opérateurs unaires ou binaires.
- #19283 DENDRIFORM (CO/BI)ALGEBRAS NEW CATEGORIES : implémente les catégories d'algèbres dendriformes, de coalgèbres codendriformes [Lod01] et de bialgèbres bi-dendriformes [Foi07].
- #19285 DIESE PRODUCT ALGEBRAS : fournit du code générique pour les #-algèbres [ANT11].
- #19286 ALGEBRAS WITH POLYNOMIAL REALIZATIONS : fournit du code générique pour les algèbres pour lesquelles il existe des réalisations polynomiales.

- [#19287](#) ALGEBRAS WITH SCALAR PRODUCT : fournit du code générique pour les structures avec un produit scalaire.

## Combinatoires

- [#13855](#) PLANAR BINARY TREES (LODAY-RONCO) HOPF ALGEBRA AND SOME BINARY TREE METHODS : implémente l'algèbre de Hopf des arbres binaires de LODAY-RONCO [[LR98](#)].
- [#14498](#) TREES AND BINARY TREES : implémente des algorithmes classiques sur les arbres binaires (lecture préfixe, infix, suffixe, rotation, *etc.*).
- [#15595](#) SHUFFLE PRODUCT : implémente un algorithme efficace du produit de mélange (utilisant un code Grey).
- [#15597](#) QUASI-SHUFFLE PRODUCT : implémente le produit de quasi-mélange.
- [#15611](#) PACKED WORDS : implémente la classe combinatoire des mots tassés.
- [#19264](#) GRADED CONNECTED HOPF ALGEBRAS DESIGN : fournit un framework d'algèbre de Hopf facile à implémenter.
- [#19278](#) FQSYM (MALVENUTO-REUTENAUER) HOPF ALGEBRAS : implémente l'algèbre de Hopf de MALVENUTO-REUTENAUER [[MR95](#)].
- [#19320](#) WQSYM (HOPF ALGEBRA OF PACKED WORDS/ORDERED SET PARTITIONS) : implémente l'algèbre de Hopf des mots quasi-symétriques [[NT06](#)].

## Autres

- [#14266](#) PRETTY CONSOLE OUTPUT -> ASCII ART : ajoute un module d'affichage AsciiArt dans le terminal.
- THÉORIE DES ESPÈCES ([u/elixyre/species](#)) Le module actuel de théorie des espèces n'est pas « friendly-user ». Ce module n'est pas utilisable (ou facilement utilisable) avec ses propres espèces. Ce code est un travail (en cours) de réécriture intégral de ce module pour intégrer la théorie des espèces au reste des structures combinatoires présentes dans SAGEMATH.

# Bibliographie

- [AL04] Marcelo AGUIAR et Jean-Louis LODAY. “Quadri-algebras”. *Journal of Pure and Applied Algebra* 191.3 (2004), p. 205–221.
- [AL15] Marcelo AGUIAR et Aaron LAUVE. “The characteristic polynomial of the Adams operators on graded connected Hopf algebras”. *Algebra & Number Theory* 9.3 (2015), p. 547–583.
- [AMR05] Marco ALMEIDA, Nelma MOREIRA et Rogério REIS. “On the representation of finite automata”. *Int. Workshop on Descriptive Complexity of Formal Systems*. 2005, p. 269–276.
- [AMR07] Marco ALMEIDA, Nelma MOREIRA et Rogério REIS. “Enumeration and generation with a string automata representation”. *Theoretical Computer Science* 387.2 (2007), p. 93–102.
- [AMR08] Marco ALMEIDA, Nelma MOREIRA et Rogério REIS. “Exact generation of minimal acyclic deterministic finite automata”. *International Journal of Foundations of Computer Science* 19.04 (2008), p. 751–765.
- [ANT11] Jean-Christophe AVAL, Jean-Christophe NOVELLI et Jean-Yves THIBON. “The  $\#$ -product in combinatorial Hopf algebras”. *23<sup>th</sup> International Conference on Formal Power Series and Algebraic Combinatorics*. Discrete Mathematics et Theoretical Computer Science. 2011, p. 75–86.
- [AV10] Jean-Christophe AVAL et Xavier VIENNOT. “The product of trees in the Loday-Ronco algebra through Catalan alternative tableaux”. *Séminaire Lotharingien de Combinatoire* 63 (2010), B63h.
- [AZ10] Martin AIGNER et Günter M. ZIEGLER. *Proofs from THE BOOK*. T. 274. Springer, 2010.
- [BDN08] Frédérique BASSINO, Julien DAVID et Cyril NICAUD. “Enumeration and random generation of possibly incomplete deterministic automata.” *Pure Mathematics and Applications* 19.2-3 (2008), p. 1–16.
- [BDS12] Frédérique BASSINO, Julien DAVID et Andrea SPORTIELLO. “Asymptotic enumeration of Minimal Automata”. *29<sup>th</sup> International Symposium on Theoretical Aspects of Computer Science*. 2012, p. 88–99.
- [Bet+14] Dan BETEA et al. “Perfect sampling algorithm for Schur processes”. [arXiv :1407.3764](https://arxiv.org/abs/1407.3764). 2014.
- [BLL98] François BERGERON, Gilbert LABELLE et Pierre LEROUX. *Combinatorial species and tree-like structures*. Cambridge University Press, 1998.

- [BN07] Frédérique BASSINO et Cyril NICAUD. “Enumeration and random generation of accessible automata”. *Theoretical Computer Science* 381.1 (2007), p. 86–104.
- [Brz62] Janusz A. BRZOWSKI. “Canonical regular expressions and minimal state graphs for definite events”. *Mathematical theory of Automata* 12.6 (1962), p. 529–561.
- [Bul+13] Jean-Paul BULTEL et al. “Redfield-pólya theorem in wsym”. *25<sup>th</sup> Formal Power Series and Algebraic Combinatorics*. 2013, p. 563–574.
- [Bur11] William BURNSIDE. *Theory of groups of finite order*. Cambridge University Press, 1911.
- [BZ09] Nantel BERGERON et Mike ZABROCKI. “The Hopf algebras of symmetric functions and quasi-symmetric functions in non-commutative variables are free and co-free”. *Journal of Algebra and Its Applications* 8.04 (2009), p. 581–600.
- [Cal08] David CALLAN. “A Determinant of Stirling Cycle Numbers Counts Unlabeled Acyclic Single-Source Automata”. *Discrete Mathematics and Theoretical Computer Science* 10.2 (2008), p. 77–86.
- [Car08] Olivier CARTON. *Langages formels, calculabilité et complexité*. T. 101. Vuibert, 2008.
- [CD12] Vincent CARNINO et Sven DE FELICE. “Sampling different kinds of acyclic automata using markov chains”. *Theoretical Computer Science* 450 (2012), p. 31–42.
- [CH04] Cezar CÂMPEANU et Wing H. Ho. “The maximum state complexity for finite languages”. *Journal of Automata, Languages and Combinatorics* 9.2-3 (2004), p. 189–202.
- [Cha09] Frédéric CHAPOTON. “Some dendriform functors”. [arXiv :0909.2751](https://arxiv.org/abs/0909.2751). 2009.
- [Cho56] Noam CHOMSKY. “Three models for the description of language”. *Information Theory, IRE Transactions on* 2.3 (1956), p. 113–124.
- [CP15] Grégory CHATEL et Vincent PILAUD. “Cambrian Hopf Algebras”. [arXiv :1411.3704](https://arxiv.org/abs/1411.3704). 2015.
- [Dev12] The Sage DEVELOPERS. *Sage Mathematics Software (Version x.y.z)*. <http://www.sagemath.org>. 2012.
- [DHT02] Gérard H. E. DUCHAMP, Florent HIVERT et Jean-Yves THIBON. “Noncommutative symmetric functions VI : Free Quasi-Symmetric functions and related algebras”. *International Journal of Algebra and Computation* 12.05 (2002), p. 671–717.
- [DKS02] Michael DOMARATZKI, Derek KISMAN et Jeffrey O. SHALLIT. “On the number of distinct languages accepted by finite automata with  $n$  states”. *Journal of Automata, Languages and Combinatorics* 7.4 (2002), p. 469–486.

- [Dom03] Michael DOMARATZKI. “Improved bounds on the number of automata accepting finite languages”. *Developments in Language Theory*. Springer. 2003, p. 209–219.
- [Dom04] Michael DOMARATZKI. “Combinatorial interpretations of a generalization of the Genocchi numbers”. *Journal of Integer Sequences* 7.04.3 (2004), p. 6–17.
- [Duc+11a] Gerard H. E. DUCHAMP et al. “Hopf algebras of diagrams”. *International Journal of Algebra and Computation* 21.06 (2011), p. 889–911.
- [Duc+11b] Gérard H. E. DUCHAMP et al. “Noncommutative symmetric functions VII : Free Quasi-Symmetric functions revisited”. *Annals of combinatorics* 15.4 (2011), p. 655–673.
- [Duc+97] Gérard H. E. DUCHAMP et al. “Noncommutative symmetric functions III : Deformations of Cauchy and convolution algebras.” *Discrete Mathematics & Theoretical Computer Science* 1.2 (1997), p. 159–216.
- [Foi07] Loïc FOISSY. “Bidendriform bialgebras, trees, and free quasi-symmetric functions”. *Journal of Pure and Applied Algebra* 209.2 (2007), p. 439–459.
- [Fom94] Sergey FOMIN. “Duality of graded graphs”. *Journal of Algebraic Combinatorics* 3.4 (1994), p. 357–404.
- [FR74] Dominique FOATA et John RIORDAN. “Mappings of acyclic and parking functions”. *Aequationes Mathematicae* 10.1 (1974), p. 10–22.
- [FS09] Philippe FLAJOLET et Robert SEDGEWICK. *Analytic combinatorics*. Cambridge University press, 2009.
- [Gel+95] Israel M. GELFAND et al. “Noncommutative Symmetrical Functions”. *Advances in Mathematics* 112.2 (1995), p. 218–348.
- [Ges84] Ira M. GESSEL. “Multipartite  $P$ -partitions and inner products of skew Schur functions”. *Contemp. Math* 34.289–301 (1984), p. 101.
- [GH96] Adriano M. GARSIA et Mark D. HAIMAN. “Some natural bigraded  $\mathcal{S}_n$ -modules and  $q, t$ -Kostka Coefficients”. *Electronic Journal of Combinatorics* 3 (1996), R24.
- [GL11] António C. H. GUEDES DE OLIVEIRA et Michel LAS VERGNAS. “Parking Functions and Labeled Trees”. *Séminaire Lotharingien de Combinatoire* 65 (2011), B65e.
- [GR91] Israel M. GELFAND et Vladimir S. RETAKH. “Determinants of matrices over noncommutative rings”. *Functional Analysis and Its Applications* 25.2 (1991), p. 91–102.
- [Hai01] Mark D. HAIMAN. “Hilbert schemes, polygraphs and the Macdonald positivity conjecture”. *Journal of the American Mathematical Society* 14.4 (2001), p. 941–1006.
- [Hai94] Mark D. HAIMAN. “Conjectures on the quotient ring by diagonal invariants”. *Journal of Algebraic Combinatorics* 3.1 (1994), p. 17–76.

- [Har60] Frank HARARY. “Unsolved problems in the enumeration of graphs”. *Institute of Mathematics, Hungarian Academy of Sciences* (1960), p. 63.
- [Hiv99] Florent HIVERT. “Combinatoire des fonctions quasi-symétriques”. *These de Doctorat, Marne-La-Vallée* (1999).
- [HMU07] John E. HOPCROFT, Rajeev MOTWANI et Jeffrey D. ULLMAN. *Introduction to Automata Theory, Languages, and Computation*. Sous la dir. de Pearson International EDITION. Addison Wesley, 2007.
- [HNT02] Florent HIVERT, Jean-Christophe NOVELLI et Jean-Yves THIBON. “Un analogue du monoïde plaxique pour les arbres binaires de recherche”. *Comptes Rendus Mathématique* 335.7 (2002), p. 577–580.
- [HNT05] Florent HIVERT, Jean-Christophe NOVELLI et Jean-Yves THIBON. “The algebra of binary search trees”. *Theoretical Computer Science* 339.1 (2005), p. 129–165.
- [Hop71] John E. HOPCROFT. “An  $n \log n$  algorithm for minimizing states in a finite automaton”. *Theory of Machines and Computations* (1971), p. 189–196.
- [HP15] Florent HIVERT et Jean-Baptiste PRIEZ. “Polynomial realization of Hopf algebras” (2015).
- [HP67] Frank HARARY et Edgar M. PALMER. “Enumeration of finite automata” (1967).
- [Kle51] Stephen C. KLEENE. “Representation of events in nerve nets and finite automata”. *Annals of Mathematics Studies* 34 (1951), p. 3–41.
- [KLT97] Daniel KROB, Bernard LECLERC et Jean-Yves THIBON. “Noncommutative symmetric functions II : Transformations of alphabets”. *International Journal of Algebra and Computation* 7.02 (1997), p. 181–264.
- [Knu70] Donald E. KNUTH. “Permutations, matrices, and generalized Young tableaux”. *Pacific Journal of Mathematics* 34.3 (1970), p. 709–727.
- [Knu81] Donald E. KNUTH. *The art of computer programming – Seminumerical Algorithms*. T. 2. Addison-Wesley Redwood, 1981.
- [Kor86] Aleksej D. KORSHUNOV. “On the number of non-isomorphic strongly connected finite automata”. *Journal of Information Processing and Cybernetics* 22.9 (1986), p. 459–462.
- [Kre80] Germain KREWERAS. “Une famille de polynômes ayant plusieurs propriétés énumératives”. *Periodica Mathematica Hungarica* 11.4 (1980).
- [KSY06] Joseph P.S. KUNG, Xinyu SUN et Catherine H. YAN. “Goncarov-Type Polynomials and Applications in Combinatorics” (2006).
- [KT97] Daniel KROB et Jean-Yves THIBON. “Noncommutative symmetric functions IV : Quantum linear groups and Hecke algebras at  $q = 0$ ”. *Journal of Algebraic Combinatorics* 6.4 (1997), p. 339–376.

- [KT99] Daniel KROB et Jean-Yves THIBON. "Noncommutative symmetric functions V : A degenerate version of  $U_q(gl_N)$ ". *International Journal of Algebra and Computation* 9.03n04 (1999), p. 405–430.
- [KW66] Alan G. KONHEIM et Benjamin WEISS. "An occupancy discipline and applications". *SIAM Journal on Applied Mathematics* 14.6 (1966), p. 1266–1274.
- [KY03] Joseph P.S. KUNG et Catherine H. YAN. "Goncarov polynomials and parking functions". *Journal of Combinatorial Theory, Series A* 102.1 (2003), p. 16–37.
- [Lis06] Valery A. LISKOVETS. "Exact enumeration of acyclic deterministic automata". *Discrete Applied Mathematics* 154.3 (2006), p. 537–551.
- [Lis69] Valery A. LISKOVETS. "The number of initially connected automata". *Kibernetika* 3 (1969), p. 16–19.
- [Lis71] Valery A. LISKOVETS. "Enumeration of non-isomorphic strongly connected automata". *Vesci Akad. Navuk BSSR, Ser. Fiz.-Mat. Navuk* 3 (1971), p. 26–30.
- [Lod01] Jean-Louis LODAY. "Dialgebras". [arXiv :math/0102053](https://arxiv.org/abs/math/0102053). 2001.
- [Loe05] Nicholas A. LOEHR. "Combinatorics of  $q, t$ -parking functions". *Advances in Applied Mathematics* 34 (2 2005).
- [LR04] Jean-Louis LODAY et Mari   O. RONCO. "Tri-algebras and families of polytopes". *Contemporary Mathematics* 346 (2004), p. 369–398.
- [LR98] Jean-Louis LODAY et Mari   O. RONCO. "Hopf algebra of the planar binary trees". *Advances in Mathematics* 139.2 (1998), p. 293–309.
- [Mac98] Ian G. MACDONALD. *Symmetric functions and Hall polynomials*. Oxford university press, 1998.
- [MM08] Manuel MAIA et Miguel M  NDEZ. "On the arithmetic product of combinatorial species". *Discrete Mathematics* 308.23 (2008), p. 5407–5427.
- [MM65] John W. MILNOR et John C. MOORE. "On the structure of Hopf algebras". *Annals of Mathematics* (1965), p. 211–264.
- [Moo56] Edward F. MOORE. "Gedanken-experiments on sequential machines". *Automata studies* 34 (1956), p. 129–153.
- [Mor82] Robert MORRIS. *Umbral calculus and Hopf algebras*. T. 6. American Mathematical Society, 1982.
- [MR95] Claudia MALVENUTO et Christophe REUTENAUER. "Duality between quasi-symmetrical functions and the solomon descent algebra". *Journal of Algebra* 177.3 (1995), p. 967–982.
- [Ner58] Anil NERODE. "Linear automaton transformations". *Proceedings of the American Mathematical Society* 9.4 (1958), p. 541–544.
- [NT04a] Jean-Christophe NOVELLI et Jean-Yves THIBON. "A Hopf algebra of parking functions". *16<sup>th</sup> Formal Power Series and Algebraic Combinatorics*. 2004, p. 215–226.



- [NT04b] Jean-Christophe NOVELLI et Jean-Yves THIBON. "Free quasi-symmetric functions of arbitrary level". [arXiv :math/0405597](https://arxiv.org/abs/math/0405597). 2004.
- [NT06] Jean-Christophe NOVELLI et Jean-Yves THIBON. "Construction of dendri-form trialgebras". *Comptes Rendus Mathematique* 342.6 (2006), p. 365–369.
- [NT07] Jean-Christophe NOVELLI et Jean-Yves THIBON. "Hopf algebras and dendri-form structures arising from parking functions". *Fundamenta Mathematicae* 193.1 (2007), p. 189–241.
- [NT08] Jean-Christophe NOVELLI et Jean-Yves THIBON. "Noncommutative symmetric functions and Lagrange inversion". *Advances in Applied Mathematics* 40.1 (2008), p. 8–35.
- [NT14] Jean-Christophe NOVELLI et Jean-Yves THIBON. "Hopf Algebras of  $m$ -permutations,  $(m + 1)$ -ary trees, and  $m$ -parking functions". [arXiv :1403.5962](https://arxiv.org/abs/1403.5962). 2014.
- [Nze06] Janvier NZEUTCHAP. "Dual graded graphs and Fomin's  $r$ -correspondences associated to the Hopf algebras of Planar Binary Trees, Quasi-symmetric Functions and Noncommutative Symmetric Functions". *18<sup>th</sup> International Conference on Formal Power Series and Algebraic Combinatorics*. Discrete Mathematics et Theoretical Computer Science. 2006, p. 632–644.
- [Per82] Daniel PERRIN. "Cours d'algèbre". *École Normale Supérieure de Jeunes Filles, Paris* 18 (1982).
- [Pit99] Jim PITMAN. "Coalescent random forests". *Journal of Combinatorial Theory, Series A* 85.2 (1999), p. 165–193.
- [Pól37] George PÓLYA. "Kombinatorische Anzahlenbestimmungen für Gruppen". *Acta Mathematica* 68 (1937), p. 145–254.
- [PR95] Stéphane POIRIER et Christophe REUTENAUER. "Algèbres de Hopf de tableaux". *Annales des sciences mathématiques du Québec* 19.1 (1995), p. 79–90.
- [Pri13] Jean-Baptiste PRIEZ. "Lattice of combinatorial Hopf algebras : binary trees with multiplicities". *25<sup>th</sup> International Conference on Formal Power Series and Algebraic Combinatorics*. Discrete Mathematics et Theoretical Computer Science. 2013, p. 1137–1148.
- [Pri15a] Jean-Baptiste PRIEZ. "Bijection : Parking-like structures and Tree-like structures". [arXiv :1503.04245](https://arxiv.org/abs/1503.04245). 2015.
- [Pri15b] Jean-Baptiste PRIEZ. "Enumeration of minimal acyclic automata via generalized parking functions". *27<sup>th</sup> International Conference on Formal Power Series and Algebraic Combinatorics*. Discrete Mathematics et Theoretical Computer Science. 2015, p. 697–708.
- [Prü18] Heinz PRÜFER. "Neuer beweis eines satzes über permutationen". *Archiv Mathematik Physical* 27 (1918), p. 742–744.



- [PS02] Jim PITMAN et Richard P. STANLEY. "A polytope related to empirical distributions, plane trees, parking functions, and the associahedron". *Discrete & Computational Geometry* 27.4 (2002), p. 603–602.
- [PV15] Jean-Baptiste PRIEZ et Aladin VIRMAUX. "Non-commutative Frobenius characteristic of generalized parking functions – Application to enumeration". *27<sup>th</sup> International Conference on Formal Power Series and Algebraic Combinatorics*. Discrete Mathematics et Theoretical Computer Science. 2015, p. 733–744.
- [Red27] John H. REDFIELD. "The theory of group-reduced distributions". *American Journal of Mathematics* (1927), p. 433–455.
- [Rio69] John RIORDAN. "Ballots and trees". *Journal of Combinatorial Theory* 6.4 (1969), p. 408–411.
- [Rob85] Robert W. ROBINSON. "Counting strongly connected finite automata". *Graph theory with applications to algorithms and computer science*. John Wiley & Sons, Inc. 1985, p. 671–685.
- [RS06] Mercedes ROSAS et Bruce SAGAN. "Symmetric functions in noncommuting variables". *Transactions of the American Mathematical Society* 358.1 (2006), p. 215–232.
- [Sch68] Marcel-Paul SCHÜTZENBERGER. "On an enumeration problem". *Journal Combinatorial Theory* 4 (1968), p. 219–221.
- [Shi08] Heesung SHIN. "A new bijection between forests and parking functions". [arXiv :0810.0427](https://arxiv.org/abs/0810.0427). 2008.
- [Shi86] Jian Yi SHI. *The Kazhdan-Lusztig cells in certain affine Weyl groups*. Springer, 1986.
- [Slo15] Neil J. A. SLOANE. *The On-Line Encyclopedia of Integer Sequences Foundation Inc.* 2015. URL : <http://oeis.org> (visité le 15/07/2015).
- [Sol76] Louis SOLOMON. "A Mackey formula in the group ring of a Coxeter group". *Journal of Algebra* 41.2 (1976), p. 255–264.
- [Sta04] Richard P. STANLEY. "An introduction to hyperplane arrangements". *Lecture notes, IAS/Park City Mathematics Institute* (2004).
- [Sta97] Richard P. STANLEY. "Parking Functions and Noncrossing Partitions". *the electronic journal of combinatorics* 4.2 (1997), R20.
- [Sta99] Richard P. STANLEY. *Enumerative combinatorics, Volume II*. T. 62. Cambridge University Press, 1999.
- [Tak71] Mitsuhiro TAKEUCHI. "Free Hopf algebras generated by coalgebras". *Journal of the Mathematical Society of Japan* 23.4 (1971), p. 561–582.
- [Wol36] Margarete C. WOLF. "Symmetric functions of non-commutative elements". *Duke Math. J.* 2.4 (1936), p. 626–637.

- [Yan00] Catherine H. YAN. "On the enumeration of generalized parking functions". *Congressus Numerantium* (2000), p. 201–210.
- [Yan01] Catherine H. YAN. "Generalized parking functions, tree inversions, and multicolored graphs". *Advances in Applied Mathematics* 27.2 (2001), p. 641–670.

# Nomenclature

## Classes combinatoires et espèces

$F[U]$	F-structures sur $U$ , 37	$Par$	espèce des partitions, 38
$\pi \prec \tau$	$\tau$ est plus grossière que $\pi$ , 20	$\mathcal{S}$	espèce des permutations, 37
$\Sigma$	alphabet, 109	$X$	espèce des singletons, 41
$\mathcal{A}, \mathcal{B}, \mathcal{C}$	alphabets, 22	$E^k$	espèce des séquences de $k$ ensembles, 49
$\tilde{\mathcal{B}}$	arbres binaires (non-étiquetés), 16	$1$	espèce un, 41
$\Omega, \Lambda$	automate, 110	$0$	espèce zéro, 41
$\mathcal{C}$	classe combinatoire abstraite, 16	$F, G$	espèces de structures abstraites, 37
$\mathcal{C}_F$	classe combinatoire des F-structures, 38	$\mathcal{F}_F^{a,b}$	espèces des $a, b$ -arborescences en F, 94
$\tilde{\mathcal{C}}_F$	classe combinatoire des types isomorphismes de F, 38	$\mathcal{F}^{a,b}$	espèces des $a, b$ -arborescences, 94
$\mathcal{C}_n$	composante graduée de $\mathcal{C}$ de degré $n$ , 16	$L$	langage, 109
$\pi^\sim$	composition conjuguée de $\pi$ , 19	$\mathcal{A}^*$	monoïde libre sur $\mathcal{A}$ , 23
$\pi$	composition, 19	$\epsilon$	mot vide, 22
$\pi \models n$	compositions de l'entier $n$ , 19	$\mathcal{A}^n$	mots sur $\mathcal{A}$ de longueur $n$ , 22
$D(\pi)$	ensemble de descentes de $\pi$ , 19	$Sh_{m,n}$	mélange des permutations de $m$ et $n$ , 24
$S, T, U$	ensemble fini, 18	$\ell$	ordre linéaire, 21
$\mathbb{X}$	ensemble infini (commutatif), 25	$\ell \ll U$	ordres linéaires de l'ensemble $U$ , 21
$\mathcal{P}_X/\mathcal{P}(\chi)$	espèce des $\chi$ -fonctions de parking, 64	$\lambda$	partition, 17
$\triangleright_{\chi}^F$	espèce des $\chi$ -tables de parking en F, 83	$\lambda \vdash U$	partitions de l'ensemble $U$ , 18
$\mathcal{B}$	espèce des arbres binaires, 44	$\lambda \vdash n$	partitions de l'entier $n$ , 17
$\mathcal{F}_{\mathcal{L}}$	espèce des arbres enracinés ordonnés, 45	$\sigma$	permutation, 21
$\mathcal{A}$	espèce des arbres enracinés, 45	$\mathcal{S}_n$	permutations de $[n]$ , 21
$a$	espèce des arbres, 88	$F_n$	restriction de F aux ensembles de cardinalité $n$ , 42
$\mathcal{C}$	espèce des cycles, 101	$F_+$	restriction de F aux ensembles non-vides, 42
$End$	espèce des endofonctions, 37	$s, t, u$	structures combinatoires, 16
$E$	espèce des ensembles, 41	$f, g$	structures d'espèce, 37
$\mathcal{P}$	espèce des fonctions de parking, 56	$(Q_i)$	séquence des ensembles $Q_i$ , 49
$\mathcal{F}$	espèce des forêts d'arbres enracinés, 45	$F[\sigma]$	transport de F-structures le long de $\sigma$ , 37
$\mathcal{L}$	espèce des ordres linéaires, 45		

## Algèbres de Hopf

$V$	$\mathbb{k}$ -espace vectoriel, 154	$Ens$	Alphabets nus, 163
$QSym$	algèbre des fonctions quasi-symétriques, 31	$\mathcal{S}$	antipode, 154
$NCSF$	algèbre des fonctions symétriques non-commutatives, 28	$\epsilon$	application counité de coalgèbre, 154
$Sym, \Lambda$	algèbre des fonctions symétriques, 25	$\iota$	application unité d'algèbre, 154
$Ens_{<}^\infty$	Alphabets nus infinis totalement ordonnés, 164	$\Delta$	coproduit de coalgèbre, 154
$Ens^\infty$	Alphabets nus infinis, 163	$\mathbb{k}$	corps, 154

$\mathbf{R}_{\pi}$	fonction (de Schur) ruban (non-commutative), 29
$\mathbf{S}_{\pi}$	fonction complète (non-commutative), 28
$h_{\lambda}$	fonction homogène, 26
$m_{\lambda}$	fonction monomiale, 25
$\mathbf{M}_{\pi}$	fonction quasi-monomiale, 32
$p_{\lambda}$	fonction somme de puissances, 25
$\Lambda_{\pi}$	fonction élémentaire (non-commutative), 28

$e_{\lambda}$	fonction élémentaire, 26
$\mu$	produit d'algèbre, 154
$f([k]_q)$	réalisation pol. de $f$ sur le $q$ -entier $k$ , 27
$f(k)$	réalisation pol. de $f$ sur $[k]$ , 27
$f(\mathbb{X})$	réalisation pol. de $f$ sur $\mathbb{X}$ , 27
$\text{sid}$	standardisation, 203
$\mathbf{Alph}$	type d'alphabet, 162

## Nombres

$\tilde{\mathcal{P}}(\chi; n)$	$\chi$ -chemins de longueur $n$ , 80
$p(\chi; n)$	$\chi$ -fonctions de parking de degré $n$ , 78
$[k]_q$	$q$ -entier $k$ , 27
$z_{\lambda}$	, 26
$\text{area}(s)$	aire du chemin (décoré) $s$ , 54
$\alpha(\pi)$	chemins d'évaluation $\pi$ sous $\chi$ , 74

$\pi(k)$	la somme partielle sur $\pi$ , 19
$b_n$	nombre de Bell, 18
$\sigma^{(k)}$	nombre de cycles de longueur $k$ de $\sigma$ , 39
$\lambda^{(k)}$	nombre de parts de valeur $k$ dans $\lambda$ , 17
$\left\{ \begin{smallmatrix} n \\ k \end{smallmatrix} \right\}$	nombre de Stirling de 2 <sup>nde</sup> espèce, 18
$\delta_{x,y}$	symbole de Kronecker, 156

## Opérateurs

$\cdot$	produit de concaténation, 23
$\sqcup$	produit de mélange, 23
$_{-}^{*}$	étoile de Kleene, 22

## Séries génératrices

$\text{ch}(\mathbb{F}; n)$	caractère de Frobenius des $\mathbb{F}$ -structures sous l'action de $\mathcal{S}_n$ , 39	$C(t)/\mathbb{F}(t)$	s.g. exponentielle de $C/\mathbb{F}$ , 16
$\mathbf{ch}(\mathbb{F}; n)$	caractère de Frobenius non-commutatif des $\mathbb{F}$ -structures sous l'action de $H_n(0)$ , 52	$\tilde{C}(t)/\tilde{\mathbb{F}}(t)$	s.g. ordinaire de $C/\mathbb{F}$ , 16
$\sigma_t(\mathbb{X})$	fonctions homogènes, 26	$\psi_t(\mathbb{X})$	sommes de puissances, 25
$\lambda_t(\mathbb{X})$	fonctions élémentaires, 26	$Z_{\mathbb{F}}$	série indicatrice de cycles de $\mathbb{F}$ , 39
$\mathcal{A}^*(t)$	monoïde libre sur l'ensemble $\mathcal{A}$ , 22		

## Autres

$\psi_k$	$k$ -translation de $\chi$ , 66	$L_q(\Omega)$	langage droit de l'état $q$ de l'automate $\Omega$ , 111
$\mathbb{K}\langle \mathcal{A} \rangle$	algèbre librement engendré par $\mathcal{A}$ , 152	$L(\Omega)$	langage reconnu par $\Omega$ , 111
$\#U$	cardinal de l'ensemble $U$ , 16	$ s $	le degré de $s$ , 16
$\sim \vee \approx$	clôture transitive de l'union de $\sim$ et $\approx$ , 24	$\ell(s)$	longueur de la séquence sous-jacente à $s$ , 17
$\text{Inv}(w)$	ensemble d'inversions de $w$ , 202	$\text{fix } g/\text{fix } \sigma$	nombre de points fixes par l'action de $g/\sigma$ , 34
$\text{Fix } g/\text{Fix } \sigma$	ensemble des points fixes par l'action de $g/\sigma$ , 34	$\Omega(q)$	partie accessible de $\Omega$ à partir de $q$ , 114
$\delta^*$	fonction de transition étendue, 110	$\mathcal{A}(w)$	restriction de $\mathcal{A}$ aux symboles de $w$ , 22
$\delta$	fonction de transition, 110	$\text{ev}(w)$	évaluation de $w$ , 207
$\chi$	fonction entière positive croissante, 64	$\text{pEv}(s)$	évaluation tassée de $s$ , 53
$\sim \wedge \approx$	intersection de $\sim$ et $\approx$ , 24		

# Index

$k$ -translation de  $\chi$ , 67

Actions de groupe, 35  
endofonctions, 38  
ensembles, 42  
espèces, 38  
fonctions de parking, 57, 58  
mots, 25  
partitions, 39  
permutations, 38  
point fixe, 35

Aire d'un chemin (décoré), 55

Algèbres de Hopf, 155  
combinatoire, 158  
tresse, 156

Algèbres libres, 153  
large, 154  
quotient commutatif, 154

Alphabet, 23, 110

Applications partitionnantes, 167  
dendriforme, 200  
standardisation, 204

Automates, 111  
accessibles, 114  
acycliques, 116  
co-accessible, 116, 136  
distingués, 113  
langages droits, 112  
langages reconnus, 112  
minimaux, 113  
non-initiaux, 112  
partie accessible, 115  
quasi-distingués, 113  
statut de terminaison, 112  
théorème de Kleene, 113

Caractère de Frobenius, 40

Caractère de Frobenius non-commutatif  
séquences de  $k$  ensembles, 53

Catégorie magmatique, 161

Catégorie monoïdale, 161

Classes combinatoires, 17  
arbre binaire (non-étiqueté), 17  
chemin décoré, 51  
contremarche, 51  
marche, 51  
chemins de Dyck, 60  
chemins de Dyck décorés

voir aussi fonctions de parking 60

composition d'entier, 20  
conjuguée, 20  
décomposition d'entier, 20, 51  
fonctions de parking, 48  
ordre linéaire, 22  
partition d'ensembles, 19  
partition d'entier, 18  
permutation, 22  
type cyclique, 40  
séquences de  $k$  ensembles, 50

Concaténation décalée des permutations,  
205

Ensemble d'inversions, 203

Ensemble de descentes d'une composition,  
20

Espèces de structures, 38  
 $\chi$ -fonction de parking voir aussi  
fonction de parking généralisée  
65

$\chi$ -table de parking, 84

$a, b$ -arborescences, 95

$a, b$ -arborescences en  $\mathbb{F}$ , 95

arbres binaires, 45

arbres enracinés ordonnés, 46

automorphisme, 39

caractère de Frobenius, 40

composition (partitionnelle), 45  
compositions, 46

- cycles, 102
- endofonctions, 38
- ensembles, 42
- fonction de parking généralisée, 65
  - contrainte de parking, 65
  - grammaire réursive, 69
- fonctions de parking, 57, 58
- forêts d'arbres enracinés, 46, 88
- isomorphisme de structures, 39
- ordres linéaires, 46
- partitions, 39, 46
- permutations, 38
- produit (de Cauchy), 44
- singletons, 42
- somme, 43
- structures, 38
- transport de structures, 38
- type d'isomorphisme, 39
- un, 42
- zéro, 42
- équation fonctionnelle, 45

Fonctions symétriques, 26

- homogènes, 27
- monomiales, 26
- sommes de puissances, 26
- élémentaires, 27

Formule de Caley, 89

- codage de Prüfer, 91
- preuve de J. PITMAN, 90
- théorème des matrices-arbres, 89

Groupe, 34

Langages rationnel, 110

Lemme/Formule de Burnside, 35

Lettre, 23

Monoïde libre, 24

Monoïde syntaxique, 111

Mots, 23

- facteur, 24
- préfixe, suffixe, 24
- sous-mot, 24
- sous-mot complémentaire, 24

Opérateurs

- produit de concaténation, 24
- produit de mélange, 24

Orbite, 35

Point de rupture d'un mot, 208

Polynôme/Fonction symétrique, 26

Relations d'équivalences, 25

- clôture transitive de l'union, 25
- homogène, 25
- intersection, 25

Stabilisateur, 35

Séries génératrices

- $\chi$ -chemins, 81
- arbres binaires (non-étiquetés), 17
- automates, 119
- automates (non-initiaux)
  - quasi-distingués, 122
- compositions d'entiers, 21
- exponentielle, 17, 39
- fonctions de parking, 62
  - équation fonctionnelle, 95
- fonctions de parking généralisées, 79
- nombre de Catalan, 63
- ordinaire, 17, 40
- ordres linéaires, 22
- partitions d'ensembles, 19
- partitions d'entiers, 18
- permutations, 41

Séries indicatrices de cycles, 40

- $\chi$ -tables de parking, 84
- ensembles, 42
- fonctions de parking, 63
- fonctions de parking généralisées, 71
- ordres linéaires, 46
- permutations, 41
- séquences de  $k$  ensembles, 52

Type d'alphabet, 163

Évaluation d'un mot, 208

Évaluation tassée, 54

## Titre : Combinatoire des fonctions de parking : Espèces, Énumération d'automates et Algèbre de Hopf

**Mots clés :** combinatoire algébrique, bijective, énumérative ; fonction de parking généralisé ; automate acyclique ; algèbre de Hopf ; fonction symétrique non-commutative ; réalisation polynomiale

Cette thèse se situe dans les domaines de la combinatoire algébrique, bijective et énumérative. Elle s'intéresse à l'étude des fonctions de parking généralisées suivant ces trois axes.

Dans une première partie, on s'intéresse aux fonctions de parking généralisées en tant qu'espèce de structures combinatoires (théorie introduite par A. JOYAL et développée F. BERGERON, G. LABELLE et P. LEROUX). On définit cette espèce à partir d'une équation fonctionnelle faisant intervenir l'espèce des séquences d'ensembles. On obtient un relèvement non-commutatif de la série indicatrice de cycles dans les fonctions symétriques non-commutatives, exprimé dans différentes bases. Par spécialisation, on obtient de nouvelles formules d'énumérations des fonctions de parking généralisées et de leurs types d'isomorphismes.

En remplaçant l'espèce des ensembles par d'autres espèces dans l'équation fonctionnelle, on définit de nouvelles structures : les  $\chi$ -tables de parking. Dans les cas particuliers où  $\chi : m \mapsto a + b(m - 1)$ , on établit une bijection entre les  $\chi$ -tables de parking et de nouvelles structures arborescentes, généralisant la bijection de C. H. YAN entre les  $\chi$ -fonctions de parking et les séquences de  $a$  forêts de  $b$ -arbres.

Dans une seconde partie, on s'intéresse à l'énumération d'automates. On commence par construire une bijection simple entre les automates (non-initiaux) et les séquences d'ensembles. À partir de cette bijection, on extrait la sous-famille des automates quasi-distingués (c'est-à-dire les automates pour lesquels les couples status de terminaison et fonction de transition des états sont distincts). L'énumération de ces automates quasi-distingués fournit

une meilleure borne supérieure pour le nombre d'automates minimaux que celle obtenue par M. DOMARATZKI & al.

Ensuite, on construit une nouvelle bijection entre les  $2m^k$ -fonctions de parking et les automates acycliques (non-initiaux) sur un alphabet à  $k$  symboles. De cette dernière, on extrait, directement sur les fonctions parking, de nombreuses informations de structure sur les automates, en particulier des informations liées à la minimalité. À partir de ces informations, on déduit une formule d'énumération des automates acycliques minimaux.

Dans une troisième partie, on formalise la technique commune de réalisation polynomiale des algèbres de Hopf : **FQSym**, **WQSym**, **PQSym**, etc.. Pour ceci, on définit la notion de type d'alphabet et d'application partitionnante. La notion d'application partitionnante formalise les bonnes propriétés de la standardisation, le tassement, la parkisation, etc. associées à ces précédentes algèbres de Hopf. On montre que certaines opérations, produit cartésien, coloration, union ou encore intersection, stabilisent ces notions. À partir de celles-ci, on définit deux constructions d'algèbres de Hopf combinatoire en dualité ; et l'on montre qu'elles sont automatiquement munies de structures d'algèbres dendrifformes et du produit  $\#$ .

En guise d'applications, on définit, pour toute famille de  $\chi$ -fonctions de parking, une application généralisant la parkisation. On montre que cette dernière est une application partitionnante si et seulement si  $\chi : n \mapsto 1 + m(n - 1)$ . Ceci permet de retrouver les algèbres de Hopf sur les  $m$ -fonctions de parking généralisées de J.-C. NOVELLI et J.-Y. THIBON.

## Title : Combinatorics around parking functions : Species, automata enumeration and Hopf algebras

**Keywords :** algebraic, bijective, enumerative combinatorics ; generalized parking functions ; acyclic automata ; Hopf algebra ; non-commutative symmetric function ; polynomial realization

This thesis comes within the scope of algebraic, bijective and enumerative combinatorics. It deals with the study of generalized parking functions following those axes.

In the first part, we are interested in generalized parking as a species of combinatorial structures (theory introduced by A. JOYAL and developed by F. BERGERON, G. LABELLE et P. LEROUX). We define this species from a functional equation involving the species of set sequences. We lift the cycle index serie to the non-commutative symmetric functions, expressed in several bases. By specialization, we obtain new enumeration formula for generalized parking and its isomorphism types.

By replacing the species of sets by others species in the functional equation, one defines new structures : the  $\chi$ -parking tables. In particular cases with  $\chi : m \mapsto a + b(m - 1)$ , we define a bijection between the  $\chi$ -parking tables and new tree structures. This gives a generalization of the C. H. YAN bijection between  $\chi$ -parking functions and the sequences of  $a$ -forest of  $b$ -trees.

In the second part, we are interested in the enumeration of automata. Firstly, we construct a simple bijection between (non-initial) automata and sequences of sets. From this bijection we ex-

tract a subfamily of quasi-distinguished automata (this means the automata which all states have distinct accepting status and transition function). We obtain a better upper bound of the number of minimal automata than the one of M. DOMARATZKI & al.

Then we construct a new bijection between  $2m^k$ -parking functions and (non-initial) acyclic automata over an alphabet of  $k$  symbols. From this bijection we extract, from parking function, informations about minimality of automata. We deduce an enumeration formula of the minimal acyclic automata.

In a third part, we formalize the common technique of polynomial realization of Hopf algebras : **FQSym**, **WQSym**, **PQSym**, etc.. We define a notion of type of alphabet and partitioning map. We highlight some operation which stabilizes these notions. Based on this, we define two constructions of dual combinatorial Hopf algebra ; and we show that they are automatically endowed of dendri-form coalgebra, and  $\#$ -product.

As an application, we define, for every family of  $\chi$ -parking functions, a generalization of the parkization. We show that this is a partitioning map if and only if  $\chi : n \mapsto 1 + m(n - 1)$ . From this we obtain the  $m$ -**PQSym** Hopf algebras of J.-C. NOVELLI et J.-Y. THIBON.